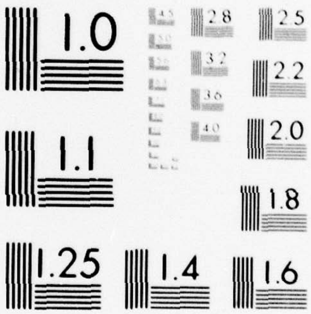
AD-A074 489 UNIVERSITY COLL LONDON (ENGLAND) DEPT OF STATISTICS --ETC F/G 17/2
UNIVERSITY COLLEGE LONDON INDRA PROJECT.(U)
MAR 79 P T KIRSTEIN N00014-77-G-0005
UNCLASSIFIED TR-56 NL

1 OF 2
AD
A074489


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

A069862

INDRA
TECHNICAL
REPORT
14 TR-56

6

UNIVERSITY COLLEGE LONDON

INDRA PROJECT.

9 ANNUAL REP~~O~~RT.

1 JAN~~UARY~~ 1~~978~~ — 31 DEC~~EMBER~~ 1978,

PROFESSOR PETER T. KIRSTEIN

10

11 MAR~~CH~~ 1979

12 144

Submitted to:

Defence Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington
Virginia 22209

Attn. Dr. Vinton G. Cerf

This research was partially supported by the US Office of Naval Research under Contract N00014-77-G-0005

15

409 714

Department of Statistics and Computer Science
University College London

LB

INDRA
TECHNICAL
REPORT
TR-56

UNIVERSITY COLLEGE LONDON

INDRA PROJECT

ANNUAL REPORT

1 JANUARY 1978 - 31 DECEMBER 1978

PROFESSOR PETER T. KIRSTEIN

MARCH 1979

Department of Statistics and Computer Science
University College London

TABLE OF CONTENTS

ABSTRACT

This report describes the activities of the UCL INDRA network group during the year 1978. Amongst the subjects covered are:

Basic implementation of the X25 network access protocol, higher level protocol above X25, network interconnection; measurements on the experimental Satnet network, facsimile transmission and Simulation of network performances.

The groups using the UCL-ARPANET link during 1978 are listed, and the INDRA participation on work on national and international standards for data communications discussed.

I. INTRODUCTION

Each year the Internetwork Display and Remote Access Group (INDRA) in the Department of Statistics and Computer Science, University College London, produces an annual report. This report is designed to summarise our work, and is a contractual commitment to many of the organisations supporting our work.

As in the previous annual report (1), this year's annual report includes not only material written specially, but also a number of papers published elsewhere. The papers included here are those most representative of our research; survey papers or bibliographies are excluded, unless they also reflect our activities.

As a result of this approach, the annual report contains very different levels of detail in the various subjects covered. In some cases only a summary is given and the references are cited; in others detailed papers are presented. Hence the length of chapters should not be considered as indicative of the relative importance of the subject. The inclusion of a paper implies only that the relevant subject has produced a reasonably finished piece of work during the year.

This year we decided that our PDP-9s would be withdrawn during 1980. As a result we have considered carefully what equipment we would require next, and how we could ensure a continuity of internal facilities and external services. We concluded that a phased changeover during the next two years was indicated - while the present generation of networks remained and the new set was becoming operational. Here it must be remembered that at least a year is needed after a data network becomes available before a significant number of computers have implemented the requisite protocols to use it. This period allows extensive interesting investigation of the network's properties, but not real service. As a result of these considerations, we have formed a plan of how our system should develop over the next couple of years, which is briefly summarised in Chapter 2.

Two themes can be singled out clearly as being important in our future research. One is a concern with the properties of the new generation of X25 networks; the second is the consideration of the problems encountered in connecting networks together. A number of UK groups are concerned with the first topic, comparatively few with the second. Several US groups in the ARPA community are concerned with the second problem, few with the first. Thus we are in a fairly unique position. We will shortly have access to many X25 networks - but are also concerned with their interconnection to satellite, local ring, and packet radio networks in addition to the more conventional technologies. Our activities on these two themes form the core of our present work, and are discussed in Chapters 3 and 4.

For several years we have been involved with a SATNET experiment. Our role has been to make User Level measurements of that network, in a way that would be relevant to other technologies. In this last year of that experiment, we have perfected our tools and made many valuable measurements. These measurements have shown up performance deficiencies, which should be remedied in 1979 as SATNET becomes an operational entity. This work is described in Chapter 5.

In 1978, the users of the UK Post Office (PO) Experimental Packet Switched Service (EPSS) held an Open Day, during which EPSS was demonstrated at 5 major sites; one of these was UCL. Throughout that day we demonstrated Network Concatenation, Measurements and SATNET altogether. This activity is discussed in Section 4.5. Network interconnections are readily possible. Thus at another Open Day at the European Information Network (EIN), both EIN and ourselves removed our customary access barriers. Users of EIN Hosts in Switzerland, Germany, France and many other countries demonstrated use of some ARPANET facilities through a concatenation of EIN, EPSS and ARPANET. It should be emphasised that such use is not possible normally. The relevant authorities on both sides of the Atlantic do not permit such usage and technical barriers have been imposed to make such usage impossible except under special approval.

An investigation of facsimile services over Data Networks has been a constant activity of the group for some years. During this year, there was a consolidation phase. Our first project terminated in the middle of the year. Most of the effort was spent in finishing off the old project and preparing for a new one which will start in 1979. Our future theme will involve much more integration of Office Automation functions with data networks and computer facilities. Our work in 1978 is described in Chapter 6.

We started an exercise several years ago of simulating specific properties of network protocols. This year, for the first time, two publications have resulted from this work; these are presented in Chapter 7. Another significant INDRA activity is the support of 'collaborative' use of ARPANET facilities between US and UK research groups. The activities in this area are discussed in Chapter 8.

The widespread use of data networks depends on easy interconnection of different computer systems and the underlying data networks. For this, agreement on a wide range of network protocols is essential. The ensuing standards activities involve many bodies with UK members - the International Consultative Committees on Telephone and

Telegraph, the International Standards Organisation, the British Standards Institute, the UK Post Office Study Groups and various EURONET groups are just some examples. The Group is deeply involved with several of these bodies. Our affiliations are listed in Chapter 9.

The key constituent of a research group is its members. In Chapter 10 the names of the group members are listed individually.

A research group like ours must be judged largely on what it produces. The products can be excellent, but are not useful if there is no publication of the results. Therefore our 1978 publications are listed in Chapter 11.

This report serves also as the annual one to a number of organisations whose support is gratefully acknowledged. These organisations are the Atomic Energy Authority (Agreements MIC 69324 and AGMT/CUL/936), the British Library (SI/G/172 and 221), the Ministry of Defence (AT/2047/064), the Science Research Council (B/RG/56168) and the US Defence Advanced Research Projects Agency via the Office of Naval Research (N00014-77-G-0005).

II. SYSTEM DEVELOPMENT

2.1 Introduction

The INDRA configuration is rapidly changing from one which, until 1977, relied heavily on a couple of DEC PDP-9s and a link to ARPANET. Over the last year the Gateway to SATNET, which is a full-sized PDP11/35 (128K words) has almost become operational. A similar sized machine has been obtained for software development and general use. Moreover, a philosophy has been developed by which dedicated LSI-11s will be used for specific functions of communication switching or protocol conversion. With this goal, we have expanded the number of LSI-11s to three, and expect four to six more during 1979.

Up to this same time our network activities have been mainly with ARPANET and EPSS. We expect to give up our direct ARPANET link in mid-1979, and our EPSS one early in 1980. The Satnet link is becoming operational now. We expect to obtain links to Euronet, IPSS and PSS in the second half of of 1979. A link to an RSRE network is expected somewhat earlier.

In view of these developments it is hardly surprising that a major activity in the future will be in techniques for network interconnection with local networks. Some of the forward thinking on this subject is discussed in Chapter 3. Our initial activities in the local network field are mentioned in Section 2.4.

2.2 The Configuration at the End of 1978

By the end of 1978, we had much more hardware than at the beginning of the year - but little more was operational from a networking standpoint. The configuration is sketched in Fig. 2.1. The link to the RSRE GEC 4080 had been removed, because the project of evaluating CORAL 66 from the US had ended. The link to the ULCC CDC 6000/7000 had been withdrawn; this was partly because ULCC needed the line, partly because the CDC 6000/7000 software was not really suited to this type of networking and partly because our PDP-9s were overloaded. Both the direct links of Culham and RL to ARPANET were permanently available, as were both the links to EPSS. These were taken down only for system development.

The Satnet connection was via a PDP-11 Gateway, and was slowly becoming operational. In addition, we had taken delivery of one 128K word PDP-11/34, with a 60M byte disk, and a total of 3 LSI-11s. The PDP-11 (actually a Systime 5000) ran UNIX. It supported both local terminals, and is planned to have asynchronous links to the LSI-11s. Software for down-line loading over the asynchronous links was under development.

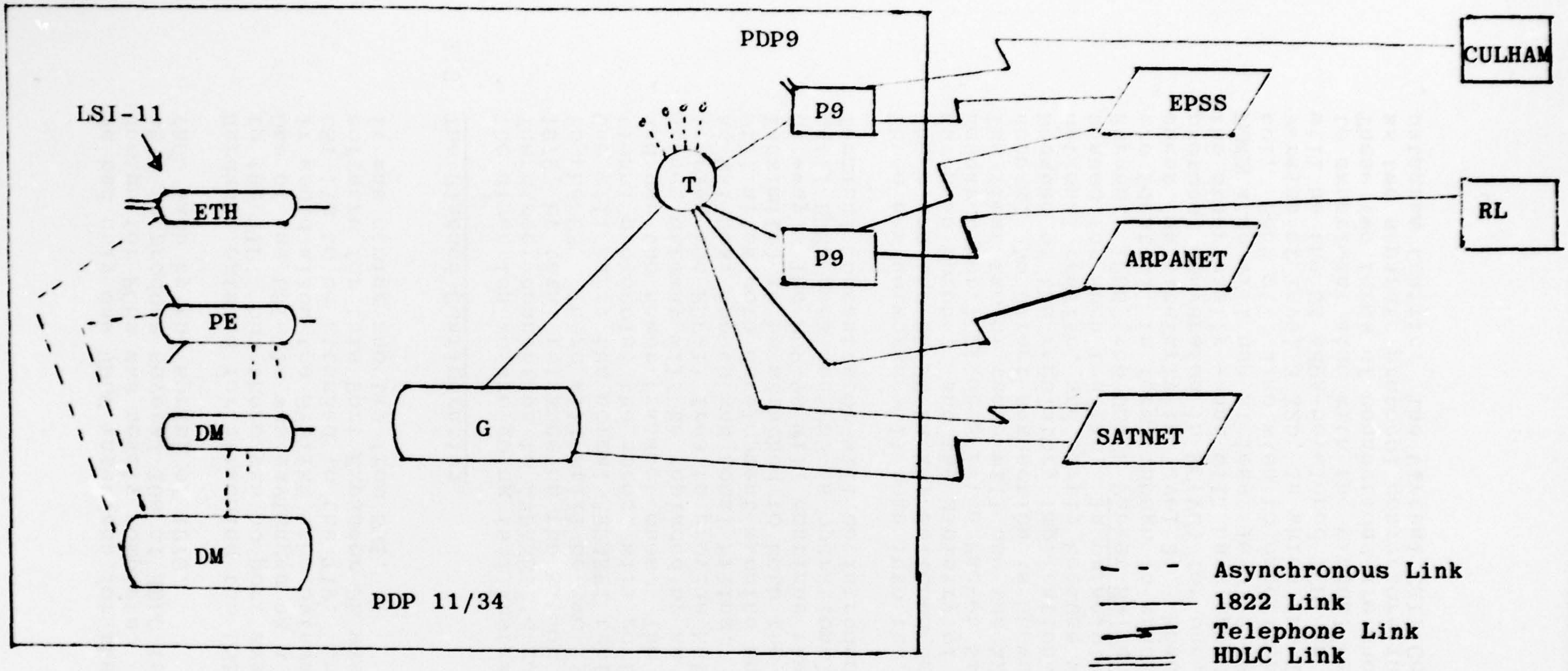


Fig. 2.1 Schematic of Configuration at the end of 1978

T = TIP
G = Gateway
PE = Port Expander
ETH = Euronet Test Host
DM = Development Machine

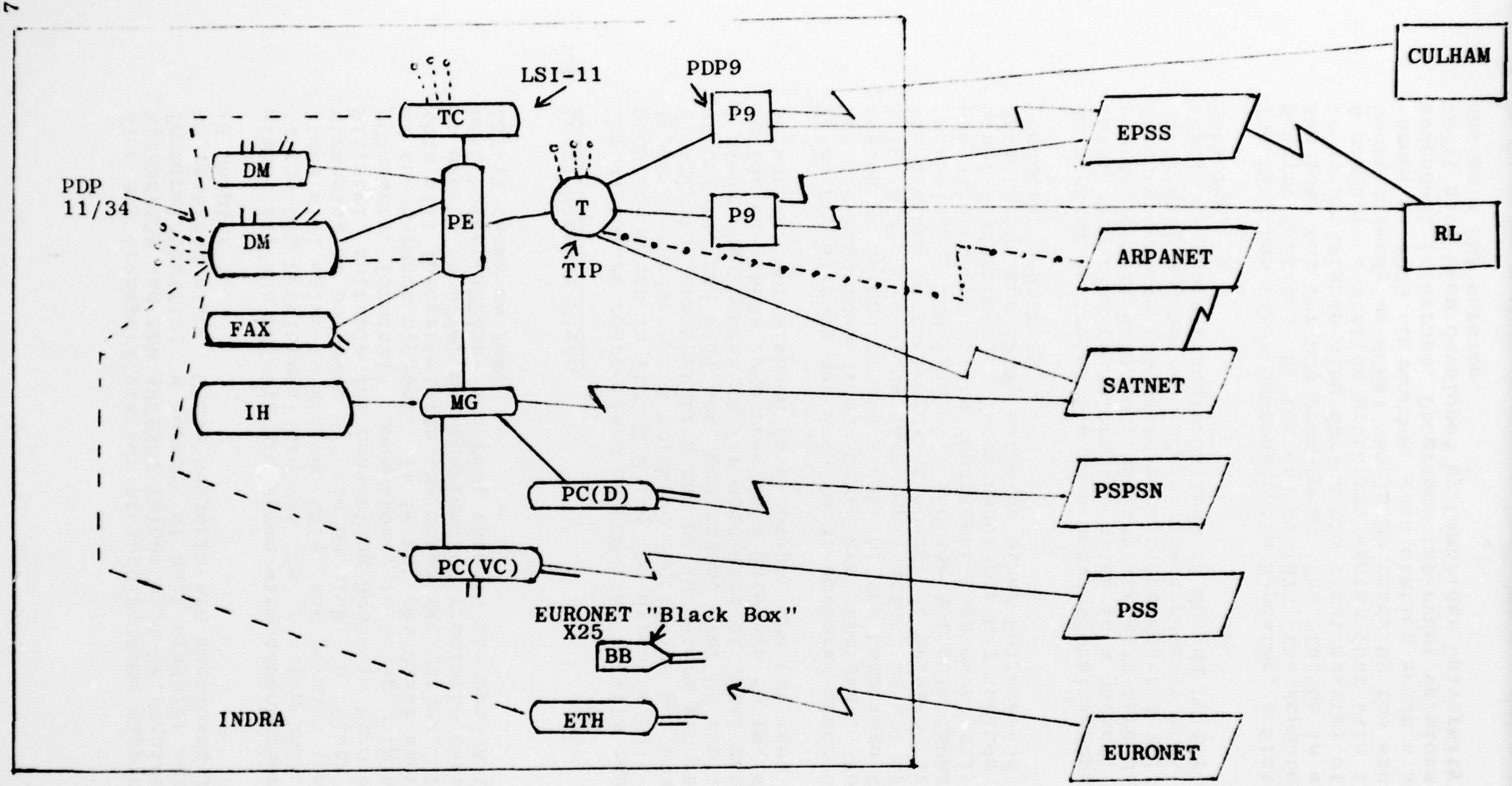
We had only one HDLC interface for the LSI-11s, but the design for PCBs was nearly complete. We expect to be able to produce several 10s of HDLC interfaces at about £500 each by the spring of 1979.

Hardware existed for attaching the LSI-11s and the SYSTIME to the TIP - but there was no port available on the TIP. One of the LSI-11s was envisaged as a TIP Port Expander; it would allow the SYSTIME, the Gateway and up to 5 LSI-11s to be attached to the TIP. Unfortunately the software for this Port Expander PE was not yet available; it was coming shortly from SRI.

2.3 The Planned Configuration

The direction we are going is illustrated in Fig. 2.2. The direct connection to ARPANET is to be replaced during 1979, by total reliance on the Satnet connection for transit to the US. There will still be two lines to the SIMP. One will carry the normal ARPANET traffic using the conventional protocols; the second will go through the Gateway and use the newer Transnet ones. The present PDP-11 Satnet Gateway will be replaced by an LSI-11 mini-Gateway, leaving the PDP-11 free to provide Network Services such as Mailbox, Speech and Local Filing. A new Port Expander will allow both development machine and services (such as Facsimile) to be attached to both the TIP and the mini-Gateway. The PDP-9s will continue their present role until EPSS goes and PSS is operational. A new LSI-11 Terminal Concentrator will be introduced.

Three new networks will come into the scene during 1979. These are the UK PO Packet Switched Service PSS, the European Euronet, and the Ministry of Defence (Royal Signal Research Est.) Pilot Service Packet Switched Network (PSPSN). The first two of these will use the X25 Virtual Call Network Access. No direct connection is shown to Euronet, partly because of the regulatory implications of switching international traffic, and partly because we do not anticipate a need for such traffic. In principle the access to Euronet, IPSS (the US-UK Packet Switched Service) and PSS are identical in form, though in practice there are differences (particularly in level 2). However, if we can do protocol conversion in PC(VC) for one, we could do it for the other easily - and will have appropriate access lines. PSPSN also will use at least level 2 X25 as an access protocol. RSRE will also wish to experiment with a datagram version of level 3 X25, in which the Transport stations will be the US ARPA-developed TCP; in addition, they expect to experiment also with the virtual call form of connections. These two forms of connections are shown diagrammatically as two separate protocol convertors for virtual call and datagram traffic. The Virtual Call Convertor PC(VC) must



- - - Asynchronous Link
——— 1822 Link
═══ HDLC Link
Telephone Link

TC = Terminal Concentrator
FAX = Facsimile Service Machine
IH = Internet Message Handler
PL = Protocol Converter
DM = Development Machine
MG = Mini Gateway

Fig. 2.2 Planned System by late 1979
(Except Direct line to ARPANET goes mid 1979)

also map transport station protocols; some mapping between X3/X28/X29 and the ARPANET Telnet will be required for the Terminal Traffic; a version of bulk transfer service will be required also. These subjects are mentioned further in Chapter 4.

Almost all the LSI-11s are shown with asynchronous lines to the UCL Development Machine (DM) running UNIX. These lines will exist to all the LSIs, and permits the bootstrapping of software off the DM disk. A "Black Box" from Sitintel, suitable for connecting Hosts to Euronet by pseudo "Character Terminal" connections, is being obtained as part of the Euronet project. It is not yet clear whether or how this will integrate with the other facilities. Finally, we expect to offer an experimental facsimile service over both Satnet-ARPANET and PSS; this is the reason for the LSI-11 shown as FAX.

2.4 Local Network Plans

The necessary connections of PDP-11s, LSI-11s and other machines shown in Fig. 2.2 are the direct result of the types of hardware and software facilities now available. Clearly the configuration has no long-term generalisable structure, and will not, one hopes, ever be replicated. A clear requirement is for some kind of local network connecting the INDRA resources with Gateways to the Wide Area Networks. Our planned developments are outlined in Fig. 2.3.

There are already may such local networks under development, and the resources of our Group are stretched. For this reason we intend to use a copy of the Cambridge Ring System (Fig. 2.3). To allow easy replication, we are arranging the designs of the relevant boards in printed circuit board technology. The ring will include not only INDRA computers, but also others in the department (and eventually perhaps other departments). The connections may include not only Satnet, PSS and EPSS, but also other College and University computing resources.

At present we are aiming at a conventional 10Mbps transfer rate. The ring transfers two bytes in a packet, so that the devices attached may use any mutually agreed protocol structure above the basic packet transport level. We have not really started specifying the necessary local protocols or their transformations in MG and PC for Wide Area Networks.

The ring has three components: a Repeater, a Station and a Monitor Station. By the end of 1978, the Repeater design had gone out for PCB prototyping. The Station is expected to be prototyped also during the first quarter of 1979. A complete 3-station prototype system (but with the components already on PCBs) should be ready by the end of the summer of 1979. In addition, each station needs a host-dependent interrupt. Programme interrupt versions for the PDP11 have been developed by Cambridge University; we will design a DMA version.

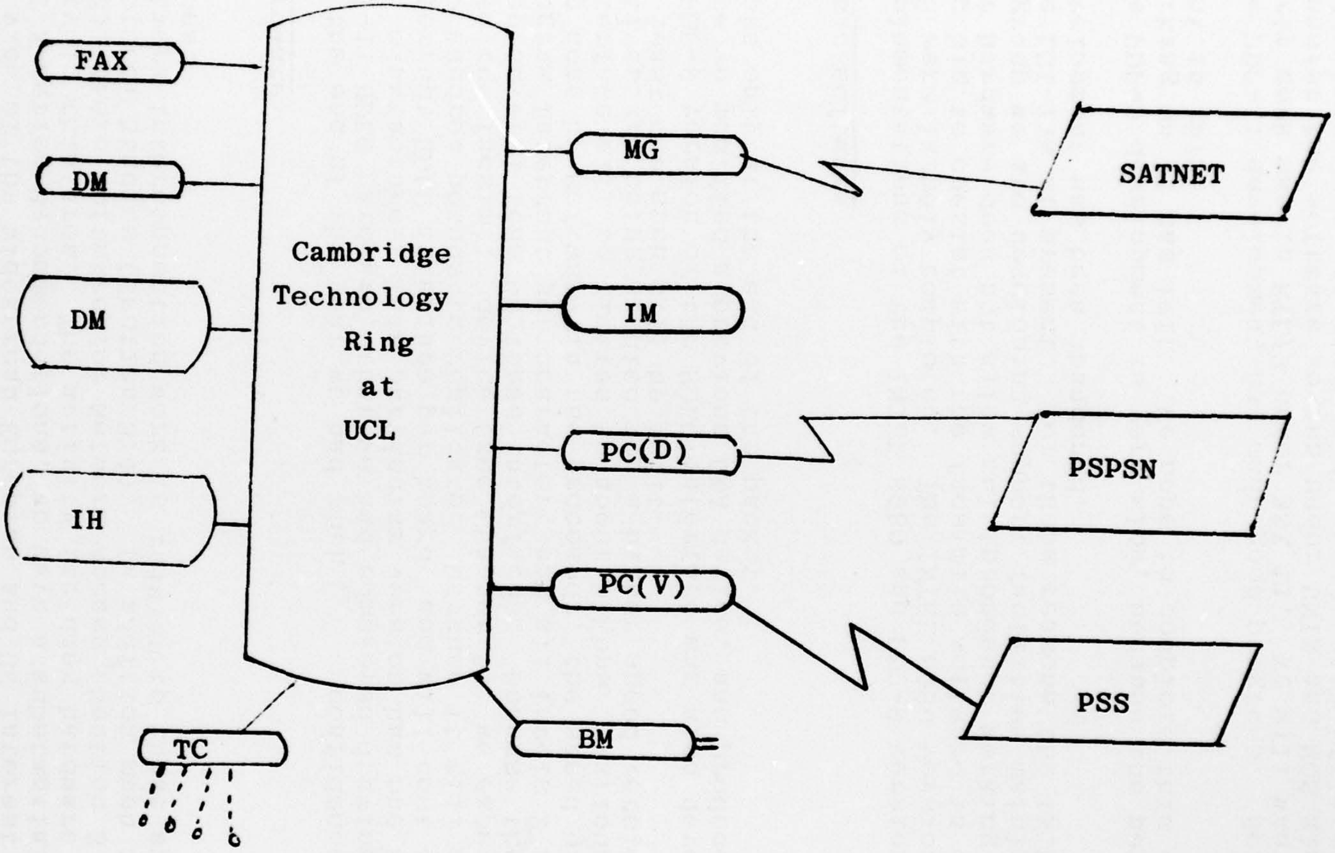


Fig. 2.3 Eventual Systems

III. X25 RELATED ACTIVITIES

3.1 Introduction

In view of the widespread European and UK interest in the PTT X-series Recommendations, we have a substantial activity in this area. The activity includes hardware (Section 3.2), basic software for Network Access (Section 3.3) and Protocol Studies (Section 3.4). In addition much of our network interconnection work is related to these specifications.

3.2 Hardware

By the end of the year we had such confidence in our LSI-11 HDLC hardware, that we had completed drawings for PCB manufacture. This particular version has one Program Interrupt HDLC interface per board, and will cost only a few hundred pounds to replicate. Because it will control also the facsimile device (see Chapter 5), we expect to need several tens of these interfaces. Because they are Program Interrupt per character, and all levels 2 and 3 are done by software in the processor, the speed of the interface will be limited to about 5 Kbps. Various alternatives are being studied for a higher speed version; at least one such will be built.

A PDP-9 version of the HDLC interface was also developed. Here we modified a previous DMA device, and produce interrupts only at the end of the packet.

3.3 Basic software

Implementations of the INTEL 8080 and PDP-9 versions of X25 were largely completed. The INTEL 8080 version proved too big to coexist with the facsimile software in one 8080. Our hardware does not allow multiprocessor working, and as a group we are developing support facilities mainly for the LSI-11s at present. For these reasons the INTEL 8080 development has been abandoned.

The PDP-9 development is very slow, because the person working on it has left. We hope to complete this development in 1979.

The PDP-11 development has made good progress. Here our early work was in RTL/2 under RSX 11. We will want to transfer the software to run under UNIX and MOS also. In order to reduce its size, and increase its efficiency, the software has been reprogrammed in Assembler, we have arranged also that it interfaces to the operating systems at only one module (COMSYS). These two factors will make it considerably easier to convert the software to run under UNIX and MOS, both of which we need for the development of Fig. 2.2. Clearly interfaces to other operating systems could be developed also. The size of both the RTL/2 and Assembly versions are shown in Table 3.1.

	RTL/2	Assembler
Frame Level (level 1)	-	½
Link Level (level 2)	4	1
Packet Level (level 3)	8	2½
COMSYS	-	½
OS Interface	-	½
TOTAL	12	5

Table 3.1 Size of Different Modules in Kwords

The level 2 has been tested successfully with the Post Office tester. The level 3 will be tested early in 1979. An overview has been written (4); both the level 2 and the Frame level of the software have been fully documented (5,6,7,8). It can be extended easily to handle multiple transmission lines (9).

3.4 X25 and High Level Protocol Studies

We have made studies of several significant problems in the protocol. The most important of these studies, how to avoid asymmetries, is presented below. In addition, we have writtem one tutorial paper (10), and commented on other X25 studies (11). We have specified a "Black Box" for connecting Hosts to X25 Networks using only their character terminal communications interfaces (12). Finally, we have made a substantial study of Remote Printing Protocols for Euronet (13); this document is being published by the Commission of the European Economic Community.

3.5 X25 Asymmetries and How to Avoid Them

X25 Asymmetries and How To Avoid Them

Colin Bradbury

Department of Statistics and Computer Science,
University College London,
Gower Street,
LONDON WC1E 6BT,
ENGLAND

INTRODUCTION

The X25 specification defines the way of connecting Data Terminal Equipment (DTE) to Data Circuit-terminating Equipment (DCE). This standard has differences between the way a DTE functions and the way a DCE functions, thus giving an asymmetrical interface. In this paper we show that it is possible to make a set of choices so that the computer software will operate in either DTE or DCE mode. Moreover, we show how the software can make decisions so that it will switch itself into the appropriate mode while remaining completely within the formal specification. We regard such software as being highly desirable for use in situations where a host on one network is a switching node on another network.

The X25 specification is divided into a number of sections. The original Link Access Procedure (LAP) has been replaced by LAPB. Although new implementations will use LAPB, older implementations are already using LAP, so we may expect that there will be a transition period in which implementations will be required which are capable of switching between LAP and LAPB. We show how this may be accomplished while remaining completely within the formal specification.

In the penultimate section of this paper, we propose a minor addition to the formal specification in order to allow error notification and recovery in a consistent manner. Whilst it is recognised that it is difficult to make changes to the formal specification at this late stage, it is also recognised that there will have to be some modifications in order to take care of existing problems [1]. We hope that the contents of this paper will enable a complete set of problems to be identified rather than having the specification updated in a piecemeal fashion as each new problem is encountered.

BACKGROUND

The INDRA research group at UCL is involved in work on several computer networks, principly Arpanet [2], Satnet [3], EPSS (The Experimental Packet Switched Service of the UK Post Office) [4], and Euronet [5]. Our activities include investigating and implementing gateway software to connect these networks together; for example, the gateway between Arpanet and EPSS is available as a service to users of these networks. It is clear that the UK successor to EPSS, which we call PSS, will be an X25 network in common with the national networks of other countries involved in Euronet. In view of this adoption of the X25 standard, we decided that, in at least one mode, our gateways should make each network appear as an X25 network to any other machine connected to the gateways.

The requirements of the gateway X25 implementations are that they communicate with DTE implementations (for applications such as facsimile terminals [6]) and with each other (for inter-network traffic); the local PSS exchange will undoubtedly expect our implementations to appear as DTEs. From these considerations, it became apparent that a gateway X25 implementation must take on the appearance expected of it by the remote station (DTE or DCE). Remote stations fall into three categories: strict DCEs which expect this station to adhere to the (implied) specification for a DTE, strict DTEs which expect this station to adhere to the formal specification for a DCE, and non-strict stations which allow this station to use either the DTE or the DCE form of the protocol. The non-strict category is subdivided into adaptive and non-adaptive stations; adaptive stations use information derived from the incoming data stream to modify their appearance, whereas non-adaptive stations always appear the same. The gateway X25 implementations are necessarily adaptive.

PROPOSED STRATEGY

The underlying strategy used in adaptive stations is first to assume that the remote station is a DCE and then modify this assumption when the remote station indicates that an error has been made; the sections below describe the modifications in detail. This strategy is followed independently at level 2 and level 3 of X25 since either of these may be used with other protocols.

As an added complication, the X25 LAP specification has been replaced by the LAPB specification. Implementations started before this change are based on LAP, while those started later are based on LAPB. It is not yet clear whether the UK PSS will provide LAP access, LAPB access, or both. Presented below are studies of the asymmetries in networks which use LAP only, networks which use LAPB only, and networks which use both LAP and LAPB. In these studies, we refer extensively to the published X25 specification [7,8,9], and the section references in brackets {} refer to sections in these documents.

LAP-ONLY NETWORKS

The only asymmetry in the LAP specification is in the content of the frame address field: DCEs use address A for commands and address B for responses, while DTEs use address B for commands and address A for responses {Section 2.4.2}.

Since the type of any particular frame is uniquely determined by the frame command byte, it is possible to ignore the address bytes of incoming frames (with the possible exception of validating that the address is either A or B). It has been suggested that this will result in an implementation which is less robust (more prone to failure) than one which uses the address field for the detection of looped communications lines. We feel that algorithms for the detection of looped lines are error prone in themselves; for example, the dropping of the low order bit of the address field may result in a station deciding that the line has become looped. We observe that a strict DCE transmitting any response frame down a looped line will consequently transmit CMDR frames ad infinitum {Section 2.4.8.2}. We also note that a line becoming looped (but undetected) may affect the higher level protocols; for example, a level 3 Call Request will always collide with itself.

For outgoing frames, this station should first assume DTE address conventions (B for commands and A for responses). If the remote station is either a strict DCE, expecting DTE address conventions, or a non-strict station, accepting either DTE or DCE address conventions, this assumption is correct. If the remote station is a strict DTE, expecting DCE address conventions, either the outgoing SARM or the UA sent in reply to the incoming SARM will cause a CMDR response from the remote station indicating invalid command (Section 2.4.8.2). On receipt of this CMDR frame, this station should switch to DCE address conventions and transmit another SARM. The remote station will retransmit its SARM after a timeout.

Bit 13 of the information field of a CMDR frame indicates whether the frame being rejected was a command or a response frame (Section 2.3.4.10). The polarity of this bit in an outgoing CMDR frame can be determined by saving the address byte of the incoming SARM for comparison with the address byte of erroneous frames.

LAPB-ONLY NETWORKS

The asymmetries in the LAPB specification are as follows:

- DCEs use address A for commands and address B for responses, while DTEs use address B for commands and address A for responses (Section 2.4.2).
- SABM is transmitted by DTEs only, although the use of SABM by DCEs is for further study (Section 2.4.5.1).
- When the link first comes up, DCEs may transmit DM frames (Sections 2.3.4.9 and 2.4.5.4.2).
- When receiving a CMDR/FRMR frame, DCEs transmit a DM response (Section 2.4.10.2 - the phrase "it is a UA or DM response" should read "it is a CMDR/FRMR or DM response").
- DCEs transmit RR, RNR, and REJ response frames, but do not transmit RR, RNR, or REJ command frames (Section ·2.3.4). This asymmetry is avoided by never transmitting RR, RNR, or REJ command frames but always responding to such frames.

As with LAP, this station should first assume DTE conventions and transmit a SABM frame containing address B; incoming DM frames should be ignored. If the remote station is a strict DCE or a non-strict station, the assumption is correct and a UA response will be returned; any incoming SABM frame should be answered with a UA response. If the remote station is a strict DTE, the outgoing SABM or the UA transmitted in response to the incoming SABM may cause the remote station to transmit a CMDR/FRMR response; on receipt of this frame, this station should switch to DCE conventions. Alternatively, the remote station may ignore both the SABM and UA frames and, after the timeout period, retransmit its own SABM; if this persists for some time (such as half of N2 timeout periods), this station should then switch to DCE conventions.

In reply to a CMDR/FRMR frame, this station should transmit a DM response. A strict DTE or non-strict station will reply to this with a SABM command; a strict DCE will reply with another DM response, to which this station should reply with a SABM command.

MIXED LAP AND LAPB NETWORKS

The initial problem here is that SARM and DM are indistinguishable until the address conventions of the remote station are known. As with LAPB, this station should first assume LAPB DTE conventions and transmit a SABM frame containing address B. If the remote station is a strict LAPB DCE or a non-strict LAPB station, the assumption is correct and a UA response will be returned (as in the LAPB case). If the remote station is a strict LAPB DTE, there will be an incoming SABM frame and communication can be established as in the LAPB case. If the remote station is a strict LAP DCE or a non-strict LAP station, the outgoing SABM frame will cause the remote station to transmit a CMDR/FRMR response; the address and information fields of this response uniquely determines what the remote station is. If the remote station is a strict LAP DTE, it will either respond to the SABM frame with a FRMR response, uniquely identifying itself, or it will continually ignore SABM frames (as invalid responses to its SARM command) and retransmit SARM commands whenever the timer runs out. If this latter situation persists for some time (such as half of N2 timeout periods), this station should switch to using DCE address conventions and transmit another SABM frame; this will cause the remote station to transmit a CMDR response as above. Transmitting a SABM frame rather than a SARM frame enables this station to adopt a common strategy for strict LAP DTEs and strict LAPB DTEs.

WINDOWING

The X25 specification states "The value of the Send State Variable is incremented by one with each successive Information frame transmission, but cannot exceed N(R) of the last received frame by more than the maximum number of outstanding frames (k)." {Section 2.3.2.4.1} where the value of k is agreed between the DTE and DCE administrations. In the absence of such an agreement, the consequences of choosing any particular value for k must be investigated.

The X25 specification originally stated that the receipt of an invalid N(S) would cause a CMDR response to be transmitted {Section 2.4.6.2}. The revisions to the specification have removed an invalid N(S) from the list of conditions for transmitting a CMDR response {Section 2.4.8.2}. We interpret this to mean that any I-frame containing an invalid N(S) should be discarded (as though the receiver was in the busy condition). By using this interpretation of the revised specification, the value of k for receiving can be set to any value up to the permitted maximum. For transmitting, the value of k should initially be set to the permitted maximum. If the remote station subsequently regards some N(S) as invalid and responds with a CMDR frame (recognisable by having zero in the third information field byte), this station should reduce the transmit value of k by one. Repeated application of this algorithm results in the transmit value of k being reduced to match the receive value of k at the remote station.

Windowing at level 3 is a similar mechanism to that at level 2, but is slightly more complex in that an N(S) error will involve a resetting of the virtual call and the possibility of consequential data loss. The only safe solution is to set the transmit value of k to one and the receive value of k to the maximum permitted value.

LEVEL 3 ASYMMETRIES

The asymmetries in the level 3 specification are as follows:

- When selecting a logical channel for an outgoing call, the DCE uses the lowest numbered logical channel which is in the ready state, while the DTE uses the highest numbered logical channel which is in the ready state {Sections 3.1.2 and 3.1.3}.
- When a call collision occurs, the DCE clears the Incoming Call and continues with the Call Request; the DTE ignores the Incoming Call packet {Section 3.1.6}. (Note that in X7x both calls are cleared).
- The DCE does not use the level 3 REJ command {Section 4.1.4}; in some networks, DTEs also may not use the level 3 REJ command. This asymmetry is avoided by not using or providing this facility.

One of the problems in connecting to a system of unknown attributes is that the number of the "highest" logical channel available is also unknown. To resolve this problem, it is necessary to define the action taken by any station when it receives a packet on a logical channel which doesn't exist (from the stations point of view). We propose that this action is specified as follows:

> "If the incoming packet is a Clear Confirmation packet, ignore it. If the incoming packet is a Clear packet (Clear Request or Clear Indication), send out a Clear Confirmation packet using the same logical channel number as used in the incoming packet. If the incoming packet is not a Clear or Clear Confirmation packet, send out a Clear packet using the same logical channel number as used in the incoming packet and with a qualifier indicating that the logical channel number is invalid."

For the general case, we partition the logical channels into two sets which we term "high" and "low"; the low set initially contains only the two lowest numbered logical channels, while the remaining logical channels are in the high set. If something odd happens on one of the low channels (such as a Call Accepted packet being received without a Call packet having been sent out), the low set is extended to include the lowest numbered logical channel from the high set which is in the ready state (and on which nothing odd has happened). The low set thus contains at least two logical channels in the ready state at both stations (if this is possible). In the following discussion, we denote by "channel 1" the lowest numbered logical channel from the low set which is in the ready state, and by "channel 2" the highest numbered logical channel from the low set which is in the ready state. The strategy requires that channel 1 and channel 2 are distinct while there are two or more logical channels in the ready state. (The low set may be defined as being all logical channels with numbers less than or equal to that of channel 2).

LEVEL 3 PROCEDURES

The initial strategy is the same as that used at level 2: assume DTE conventions until it is known what the remote station is. This station exists in one of four modes: provisional DTE (the initial mode), provisional DCE, confirmed DTE, and confirmed DCE; mode changes which can occur are from either provisional mode to any other mode (these changes are described below). The first call to be made gives us three cases to consider: an outgoing call, an incoming call, or a call collision.

An incoming call will be on either channel 1 or some other logical channel in the ready state (unless there is only one free logical channel left due to outgoing calls); this determines what the remote station is. This station should switch to confirmed DCE mode or confirmed DTE mode as appropriate. If the incoming call uses the only remaining free logical channel, nothing can be determined about the remote station so no mode change occurs and the next call to be made is again regarded as the first call.

An outgoing call should use the lowest numbered logical channel from the high set which is in the ready state. If the logical channel number is valid, this call will be acceptable regardless of what the remote station is (since the remote station cannot determine the state of the logical channels as seen by this station). If the call is cleared with an "invalid channel" qualifier, this channel and all others with higher numbers should be removed from the high set and marked "not to be used"; the call can then be reattempted on another logical channel. If there are no logical channels other than channel 1 and channel 2 in the ready state, the outgoing call should use channel 2 if this is in the ready state and channel 1 otherwise (if possible). No mode changes occur at this station because of this call, and the next call to be made (by either station) is again regarded as the first call.

If a call collision occurs on the only available logical channel, this station should clear its outgoing call and continue with the incoming call; this action is necessary since the remote station may be a strict DTE. (An alternative is to clear both calls as in X7x). Note that if both stations take this DCE action, the channel will be cleared when either station transmits a Call Accepted packet. If this station receives a Call Accepted packet before transmitting its Call Accepted packet, it should clear both calls and change to confirmed DTE mode (since the remote station has DCE potential). If the outgoing Call Accepted packet does not result in the call being cleared with a "procedure error" qualifier, the remote station is a DTE so this station should change to confirmed DCE mode. Some implementations may be able to defer clearing the outgoing call until the incoming call is either accepted or refused; for two adaptive stations, this sets up a race condition on the Call Accepted packets which could resolve the conflict, particularly when one of the stations is an inter-network gateway.

If a call collision occurs on a logical channel other than channel 1 and outgoing calls have previously been attempted, the remote station is known to be a strict DTE so this station should immediately switch to confirmed DCE mode and act accordingly.

The only case left to consider is that of a call collision when no previous calls have been made. As in the case of a call collision on the only available logical channel, this station should clear its outgoing call and continue with the incoming call. Here, however, this station can base its future strategy on the contents of the call packets by doing a byte-by-byte comparison and using the first pair of dissimilar bytes to make a decision: if the byte from the incoming packet has higher value (byte values are in the range 0 to 255) switch to provisional DCE mode and use channel 1 for the next outgoing call, whereas if the byte from the outgoing packet has higher value switch to provisional DTE mode. If the packets are identical, this station should remain in its current mode until more significant information arrives. Comparing the whole packets is rather better than comparing, for example, the DTE address fields since the contents of the latter usually depend on the X25 implementations whereas the contents of the Call User Data Field depends on the applications making the calls. Note that an arbitrary decision has been made on the direction of the test; the remote station may use different criteria, resulting in the race condition not being cleared despite the fact that this station was able to make a decision.

CONCLUSIONS

This study shows that there is a "universal" X25 implementation which is directly compatable with all X25 implementations (including itself). Moreover, a complete set of choices has been proposed to ensure that a specific implementation has this feature. In order to make level 3 error notification complete and consistent, we have proposed an extension to the protocol to deal with the case where the logical channel number is considered invalid.

REFERENCES

1 B.Cosell. Letter to the editor, Computer Communication Review, vol.8 no.2 p7; April 1978.

2 L.G.Roberts and B.D.Wessler. The ARPA computer network, Computer Communications Networks, Prentice Hall, pp485-499, 1973.

3 S.W.Treadwell, A.J.Hinchley, and C.J.Bennett. A high level network measurement tool, proc. Eurocomp 78, pp35-49, 1978.

4 P.L.Higginson and Z.Z.Fisher. Experiences with the initial EPSS service, proc. Eurocomp 78, pp581-600, 1978.

5 B.Duntester. Network control and monitoring for an international public network (Euronet), proc. Eurocomp 78, pp985-991, 1978.

6 S.Yilmaz and P.T.Kirstein. UCL experiments in facsimile transmission using database management facilities on ARPANET, proc. Eurocomp 78, pp789-820, 1978.

7 Draft recommendation X25, CCITT AP VI-No. 55-E, Geneva 1976.

8 Rapporteurs report, CCITT COM VII-No. 100-E, Geneva April 1977.

9 Rapporteurs report, CCITT COM VII-No. 123-E, Geneva October 1977.

IV. NETWORK INTERCONNECTION

4.1 Introduction

Clearly a large part of our work is concerned with network interconnection. In order to develop the configuration plan of Section 2.3, we have gone through many iterations (14-17). It would be mainly of historical interest to trace through why we are proposing now the configuration of Section 2.3.

Our work has had many aspects. Several papers (e.g. 18) discussed the connection of campus networks to public networks. In a more major published paper, we collaborated with V.G. Cerf in giving a general review of the issues in interconnecting packet networks. This paper is reproduced as Section 4.2. In another paper, we collaborated with G.R. Grossman in analysing the issues raised by the PTT X25 recommendation for the connection of Public Packet Data Networks; this paper is reproduced as Section 4.3. Another contribution (with C. Sunshine) (19) was the preparation of an IFIP document on the same subject to the CCITT.

High level protocols over concatenated networks have concerned us greatly. We have put in a major effort in trying to define the services that are required (20) and then setting out to meet the requirements. A key problem here is the provision of bulk data transfer facilities. One activity here has been the mapping, on a PDP-9, between the ARPANET and EPSS methods of operation. This development is well advanced (21), and should be completed in 1979. A much more significant development has been to start implementing a Network Independent File Transfer Protocol (NIFTP) on an ARPANET Host, in order to stage file transfers over concatenated Virtual Calls. Some details of specific applications are given in Working Papers (22). A more general discussion of the problems involved is given as Section 4.4. We propose to evaluate the differences between the protocol translation of (21) and the file staging of Section 4.4.

In June there was an EPSS open day at UCL; as part of this we demonstrated Satnet experiments through a concatenation of ARPANET, EPSS and Satnet. A discussion of that demonstration, which can be made at any time since it uses our standard service offerings - is given in Section 4.5.

4.2 Issues in Packet-Network Interconnection

Issues in Packet-Network Interconnection

VINTON G. CERF AND PETER T. KIRSTEIN

Invited Paper

Abstract—This paper introduces the wide range of technical, legal, and political issues associated with the interconnection of packet-switched data communication networks. Motivations for interconnection are given, desired user services are described, and a range of technical choices for achieving interconnection are compared. Issues such as the level of interconnection, the role of gateways, naming and addressing, flow and congestion control, accounting and access control, and basic internet services are discussed in detail. The CCITT X.25/X.75 packet-network interface recommendations are evaluated in terms of their applicability to network interconnection. Alternatives such as datagram operation and general host gateways are compared with the virtual circuit methods. Some observations on the regulatory aspects of interconnection are offered and the paper concludes with a statement of open research problems and some tentative conclusions.

I. Introduction

IT IS THE THEME of many papers in this issue, that people need access to data resources. In many cases this access must be over large distances, in others it may be local to a building or a single site. Data networks have been set up to meet many user needs—often, but not necessarily, using packet-switching technology. For single organizations, these data networks are often private ones, built with a technology optimized to the specific application. For communication between organizations, these networks are being set up by licensed carriers. In North America, there are many such licensed carriers, e.g., TELENET [1], DATAPAC [2], and TYMNET [3]. In the rest of the world, the Post, Telegraph, and Telephone Authority (PTT) in each country has a near monopoly on such services; special public data networks being set up in these countries include TRANSPAC [5] in France, EURONET [6] for inter-European traffic, DDX [7] in Japan, EDS [8] in the Federal Republic of Germany, and the Nordic Public Data Network (NPDN, [9]) in Scandinavia. These public data networks are considered in greater detail in other references (e.g., [10]-[12]). Most of the above networks use packet-switching technology; some of them, e.g., EDS and the NPDN, do not do so yet, but may do so in the future. In some cases special data networks have been authorized for specific communities, e.g., SITA [13] for the airlines, and SWIFT [14] for the banks. In addition many private networks have been set up among individual organizations, and experimental networks of different technologies have been developed also, e.g., ARPANET [15], [16], CYCLADES [17], ETHERNET [18], SPYDER [19], PRNET [20], [21] and SATNET [22].

Manuscript received June 20, 1978; revised July 21, 1978.

V. G. Cerf is with the Advanced Research Projects Agency, U.S. Department of Defense, Arlington, VA 22209.

P. T. Kirstein is with the Department of Statistic and Computer Science, University College, London, England.

It is a common user requirement that a single terminal and access port should be able to access any computing resource the user may desire—even if the resource is on another data network. From this requirement, there is a clear user need to have data networks connected together. By the same token, the providers of data network services would like to have their networks used as intensively as possible; thus they also have a strong motivation to connect their data networks to others. As a result of these considerations, there has been a high recent interest in the issues arising in the connection of data networks [23]-[26], [32].

From the user viewpoint, the requirement for interconnection of data networks is independent of the network technology. From the implementation viewpoint, there can be some considerable complications in connecting networks of widely different technologies—such as circuit-switched and datagram packet-switched networks (these terms are explained below). On the whole we will consider only, in this paper, the interconnection of packet-switched data networks. In many cases, however, the arguments will be equally valid for the interconnection of packet-switched to circuit-switched networks.

Network interconnection raises a great many technical, legal, and political questions and issues. The technical issues generally revolve around mechanisms for achieving interconnection and their performance. How can networks be interconnected so that packets can flow in a controllable way from one net to another? Should all computer systems on all nets be able to communicate with each other? How can this be achieved? What kind of performance can be achieved with a set of interconnected networks of widely varying internal design and operating characteristics? How are terminals to be given access to resources in other networks? What protocols are required to achieve this? Should the protocols of one net be translated into those of another, or should common protocols be defined? What kinds of communication protocol standards are needed to support efficient and useful interconnection? Who should take responsibility for setting standards?

The legal and political issues are at least as complex as the technical ones. Can private networks interconnect to each other or must they do so through the mediation of a public network? How is privacy to be protected? Should there be control over the kinds of data which move from one net to another? Are there international agreements and conventions which might be affected by international interconnection of data networks? What kinds of charging and accounting policies should apply to multinetwork traffic? How can faults and errors be diagnosed in a multinet environment? Who should be responsible for correcting such faults? Who should be responsible for maintaining the gateways which connect nets together?

We cannot possibly answer all of these questions in this paper, but we deal with many of them in the sections below.

This paper is divided into eleven sections. In the next section we provide some definitions, and in Section III we explore some of the motivations for network interconnection. In Section IV we discuss the range of end-user service requirements and choices for providing multinetwork service. Section V reviews the concept of computer-communication protocol layering. Section VI reviews the basic interconnection choices and introduces the concept of gateways between nets, protocol translation and the impact of common protocols; it elaborates also on the function of gateways. Section VII discusses the CCITT recommendations X.25 and X.75 and their role in network interconnection. Section VIII describes some of the network interconnections achieved and some of the experiments in progress. Section IX outlines regulatory issues raised by network interconnection alternatives. Section X mentions some unresolved research questions, and the final section offers some tentative conclusions on network interconnection issues.

II. The Definition of Terms

The vocabulary of networking is extensive and not always consistent. We introduce some generic terms below which we will use in this paper for purposes of discussion. It is important for the reader not to make any *a priori* assumptions about the physical realization of the objects named or of the boundary of jurisdictions owning or managing them. For instance, a gateway (see below) might be implemented to share the hardware of a packet switch and be owned by a packet-switching service carrier; alternatively it might be embedded in a host computer which subscribes to service on two or more computer networks. Roughly speaking, we are assigning names to groups of functions which may or may not be realized as physically distinct entities.

Packet: A packet of information is a finite sequence of bits, divided into a control header part and a data part. The header will contain enough information for the packet to be routed to its destination. There will usually be some checks on each such packet, so that any switch through which the packet passes may exercise error control. Packets are generally associated with internal packet-network operation and are not necessarily visible to host computers attached to the network.

Datagram: A finite length packet of data together with destination host address information (and, usually, source address) which can be exchanged in its entirety between hosts, independent of all other datagrams sent through a packet switched network. Typically, the maximum length of a datagram lies between 1000 and 8000 bits.

Gateway: The collection of hardware and software required to effect the interconnection of two or more data networks, enabling the passage of user data from one to another.

Host: The collection of hardware and software which utilizes the basic packet-switching service to support end-to-end interprocess communication and user services.

Packet Switch: The collection of hardware and software resources which implements all intranetwork procedures such as routing, resource allocation, and error control and provides access to network packet-switching services through a host/network interface.

Protocol: A set of communication conventions, including formats and procedures which allow two or more end points to communicate. The end points may be packet switches, hosts, terminals, people, file systems, etc.

Protocol Translator: A collection of software, and possibly hardware, required to convert the high level protocols used in one network to those used in another.

Terminal: A collection of hardware and possibly software which may be as simple as a character-mode teletype or as complex as a full scale computer system. As terminals increase in capability, the distinction between "host" and "terminal" may become a matter of nomenclature without technical substance.

Virtual Circuit: A logical channel between source and destination packet switches in a packet-switched network. A virtual circuit requires some form of "setup" which may or may not be visible to the subscriber. Packets sent on a virtual circuit are delivered in the order sent, but with varying delay.

PTT: Technically PTT stands for Post, Telegraph, and Telephone Authority; this authority has a different form in different countries. In this paper, by PTT we mean merely the authority (or authorities) licensed in each country to offer public data transmission services.

We have attempted to make these definitions as noncontroversial as possible. For example, in the definition of packet switch, we alluded to a host/network interface. The reader should not assume that subscriber services are limited to those offered through the host/network interface. The packet-switching carrier might also offer host-based services and terminal access mechanisms as additional subscriber services.

III. The Motivating Forces in the Interconnection of Data Networks

In the introduction, we mentioned that there was a strong interest, among both the users and suppliers of data serivces, in the interconnection of data networks. However, the technical interests of the different parties are not identical. The end user would merely like to be able to access any resources from a single terminal, with a single access port, as economically as possible according to his own performance criteria. A Public Carrier, or PTT, has a strong motivation to connect its network to other PTT's. As in the telephone system, the concept of all subscribers being accessible through a single Public Data Service, is considered highly desirable; however the different PTT's may have restricted geographic coverage, or only a specific market penetration.

The motivation of the PTT's to interface to private networks is weaker and more complex. They always provide facilities to attach single terminals, where a terminal may be a complex computer system; they are often not interested, at present, in making any special arrangements when the "terminal" is a whole computer network. The operators of private networks often have a vital interest in connecting their networks to other private networks and to the public ones. Even though in many cases the bulk of its traffic is internal to the private network, which is why it was set up in the first place, there is usually a vital need to access resources not available on that network. The regulatory limitations often imposed on the method of interconnection of private networks are discussed in Section IX. In some countries, it is not permitted to build private networks using leased line services, but intrabuilding networks may be permitted. Interconnection of such local networks to public networks may play a crucial role in making the local network useful.

To date the PTT's have tried to standardize on access procedures for their Public-Packet Data Services. The standardization has taken place in the International Consultative Committee on Telegraphy and Telephony (called CCITT) in a set of recommendations called X.3, X.25, X.28, and X.29 ([27]-[29]). Not all PTT's have such forms of access yet, but most of the industrialized nations in the West are moving in this direction. This series of recommendations is discussed in much more detail in Section VI; it does not pay special attention to the attachment of private networks ([31], [32]), but the recommendations are themselves expected to change to meet this requirement. The PTT's are agreeing on a set of interface recommendations and procedures called X.75 [33], to connect their networks to each other; so far this interface procedure (and its corresponding hardware) is not intended to be provided to private networks.

While most PTT's have preferred to ignore the technical implications of the attachment of private networks to the public ones, most private network operators cannot ignore this requirement. They are often motivated to add some extra "Foreign Exchange" capability as an afterthought, with minimum change to their intranetwork procedures; this approach can be successful up to a point, but will usually be limited by the lack of high-level procedures between the different networks. These high-level procedures have not yet been considered by CCITT, but it has been proposed that CCITT Study Group VII investigate high-level procedures and architectural models, in cooperation with the investigation of "open system architectures" by Technical Committee 97, Sub-Committee 16 of the International Standards Organisation (ISO). This subject is also considered later in this paper, in Section VI.

An aim of these standardization exercises is to ensure that both manufacturer and user implementations of network resources can communicate with each other through single private or public data networks. A consequence should be that the resources are also compatibly accessible over connected data networks.

Depending on the applications and spatial distribution of subscribers, the preferred choice of packet-switching medium will vary. Intrabuilding applications such as electronic office services may be most economically provided through the use of a coaxial-packet cable system such as the Xerox ETHERNET [18] and LCSNET [64], or twisted pair rings such as DCS [34], coupled with a mix of self-contained user computers (e.g., intelligent terminals with substantial computing and memory capacity) and shared computing, storage, and input-output facilities. Larger area regional applications might best employ shared video cables [35] or packet radios [20], [21] for mobile use. National systems might be composed of a mixture of domestic satellite channels and conventional leased-line services. International systems might use point-to-point links plus a shared communication satellite channel and multiple ground stations to achieve the most cost-effective service.

A consequence of the wide range of technologies which are optimum for different packet-switching applications is that many different networks, both private and public, may co-exist. A network interconnection strategy, if properly designed, will permit local networks to be optimized without sacrificing the possibility of providing effective internetwork services. The potential economic and functional advantages of local networks such as ETHERNET or DCS will lead naturally to private user networks. Such private network developments are analogous to telephone network private automated branch exchanges (PABX) and represent a natural consequence of the marriage of computer and telecommunication technology.

Two further developments can be expected. First, organizations which are dispersed geographically, nationally, or internationally, will want to interconnect these private networks both to share centralized resources and to effect intraorganization electronic mail and other automated office services. Second, there will be an increasing interest in interorganization interconnections to allow automated procurement and financial transaction services, for example, to be applied to interorganization affairs.

In most countries where private networks are permitted, interorganization telecommunication requires the involvement of a PTT. Hence the most typical network interconnection scenarios will involve three or four networks. Within one national administration the private nets of different organizations will be interconnected through a public network. International interconnections will involve at least two public networks. We will return to this topic in Section VI.

In addition to permitting locally optimized networks to be interconnected, a network interconnection strategy should also support the gradual introduction of new networking technology into existing systems without requiring simultaneous global change throughout. This consideration leads to the conclusion that the public data networks should support the most important user requirements for internet service from the outset. If this were the case, then changes in network technology which require a multinetwork system during phased transition would not, *a priori*, have to affect user services.

IV. Provision of End-User Multinetwork Services

The ultimate choice of a network interconnection strategy will be strongly affected by the types of user services which must be supported. It is useful to consider the range of existing and foreseeable user service requirements without regard for the precise means by which these requirements are to be met. We will leave for discussion in subsequent sections the choice of supporting the various services within or external to the packet-switched network. The types of service discussed below are general requirements for network facilities. For this reason they also should be supported across interconnected networks.

Most of the currently prevalent computer-communication services fall into four categories:

1) terminal access to time-shared host computers;
2) remote job entry services (RJE);
3) bulk data transfer;
4) transaction processing.

The time-sharing and transaction services typically demand short network and host response times but modest bandwidth. The RJE and file transfer services more often require high amounts of data transfer, but can tolerate longer delay. Some networks were designed to support primarily terminal service, leaving RJE or file transfer services to be supported by dedicated leased lines. Packet-switching techniques permit both types of service to be supported with common network resources, leading to verifiable economies. However, bulk data transfer requires increasingly higher throughput rates if delivery delays are to be kept constant as the amount of data to be transferred increases.

As distributed operating systems become more prevalent, there will be an increased need for host-to-host transaction services. A prototypical example of such a system is found in the DARPA National Software Works [4], [36]. In such a system, small quantities of control information must be exchanged quickly to coordinate the activity of the distributed components. Broadcast or multidestination services will be needed to support distributed file systems in which information can be stored redundantly to improve the reliability of access and to protect against catastrophic failures.

Transaction services are also finding application in reservation systems, credit verification, point of sale, and electronic funds-transfer systems in which hundreds or thousands of terminals supply to, or request of, hosts small amounts of information at random intervals. Real-time data collection for weather analysis, ground and air traffic control, and meter reading, for example, also fall into this category.

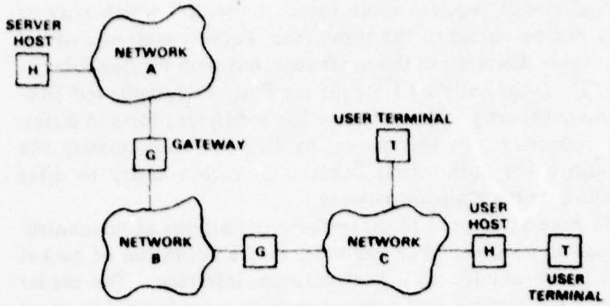


Fig. 1. Network concatenation.

More elaborate user requirements can be foreseen as electronic mail facilities propagate. Multiple destination addressing and end-to-end encryption for the protection of privacy as well as support for text, digitized voice, and facsimile message transmission are all likely requirements. Electronic teleconferencing using mixtures of compressed digital packet speech, videographics, real-time cursors (for pointing at video images under discussion), and text display will give rise to requirements for closed user groups and time-synchronized mixes of transaction-like (e.g., for cursor tracking and packet speech) and reliable circuit-like services (e.g., for display management).

Reliability and rapid response will be increasingly important as more and more computer-based applications requiring telecommunications are integrated into the business, government, military, and social fabric of the world economy. The more such systems are incorporated into their daily activities, the more vulnerable the subscribers are to failures. Reliability concerns lead to the requirement for redundant alternatives such as distributed file systems, richly connected networks, and substantial local processing and storage capability. These trends increase the need for networking to share common hardware and software resources (and thus reduce their marginal cost), to support remote software maintenance and debugging, and to support intra- and inter-organizational information exchange.

We have described the end-user services required across one or more data networks. We have carefully refrained from discussing which services should be provided in the data network, and which should be provided in the hosts. Here the choice in single networks will depend on the network technology and the application requirements. For example, in a network using a broadcast technology such as ETHERNET or the SATNET, multidestination facilities may well be incorporated in the data network itself. In typical store-and-forward networks, this feature might be provided at the host level by the transmission of multiple copies of packets. This example highlights immediately the difficulty of using sophisticated services at the data network level across concatenated networks. If *A*, *B*, and *C* are data networks connected as in Fig. 1, and *A* and *C* but not *B* support broadcast or real-time features, it is very difficult to provide them across the concatenation of *A*, *B*, and *C*.

The problem of achieving a useful set of internetwork services might be approached in several ways, as follows.

1) Require all networks to implement the entire range of desired services (e.g., datagram, virtual circuit, broadcast, real-

time, etc.), and then attempt to support these services across the gateways between the networks.

2) Require all networks to implement only the most basic services (e.g., datagram or virtual circuit), support these services across gateways, and rely on the subscriber to implement all other services end-to-end.

3) Allow the subscriber to identify the services which he desires and provide error indications if the networks involved, or the gateways between them, cannot provide the desired services.

4) Allow the subscriber to specify the internetwork route to be followed and depend on the subscriber to decide which concatenation of services are appropriate and what end-to-end protocols are needed to achieve the ultimately preferred class of service.

5) Provide one set of services for local use within each network and another, possibly different set for internetwork use.

The five choices above are by no means exhaustive, and, in fact, only scratch the surface of possibilities. Nothing has been said, thus far, about the compatibility of various levels of communication protocols which exist within each network, within subscriber equipments, and within the logical gateway between networks. To explore these issues further, it will be helpful to have a model of internetwork architecture, taking into account the common principle of protocol layering and the various possible choices of interconnection strategy which depend upon the protocol layer at which the networks are interfaced. We consider this in the next section.

V. Layered Protocol Concepts

Both to provide services in single networks, and to compare the capabilities of different networks, a very useful concept in networking is protocol layering. Various services of increasing capability can be built one on top of the other, each using the facilities of the service layer below and supporting the facilities of the layer above. A thorough tutorial on this concept can be found in the paper by Pouzin and Zimmermann in this issue [37]. We give some specific examples below of layering as a means of illustrating the scope of services and interfaces to be found in packet networks today—and some of the problems encountered in offering services across multiple networks.

Table I offers a very generic view of a typical protocol hierarchy in a store-and-forward computer network, including layers usually found outside of the communication network itself. There are several complications to the use of generic protocol layering to study network interconnection issues. Chief among these is that networks do not all contain the same elements of the generic hierarchy. A second complication is that some networks implement service functions at different protocol layers. For instance, virtual circuit networks implement an end/end subscriber virtual circuit in their intranet, end/end level protocol. Finally, the hierarchical ordering of functions is not always the same in all networks. For instance, TYMNET places a terminal handling protocol within the network access layer, so that hosts look to each other like one or more terminals. Figs. 2–7 illustrate the functional layering of some different networks. It is important to note how the functions vary with the choice of transmission medium.

TABLE I
Generic Protocol Layers

PROTOCOL LAYER	FUNCTIONS
6 APPLICATION	FUNDS TRANSFER, INFORMATION RETRIEVAL, ELECTRONIC MAIL, TEXT EDITING
5 UTILITY	FILE TRANSFER, VIRTUAL TERMINAL SUPPORT
4 END/END SUBSCRIBER	INTERPROCESS COMMUNICATION (E.G. VIRTUAL CIRCUIT, DATAGRAM, REAL TIME, BROADCAST)
3 NETWORK ACCESS	NETWORK ACCESS SERVICES (E.G. VIRTUAL CIRCUIT, DATAGRAM ...)
2 INTRANET, END TO END	FLOW CONTROL, SEQUENCING
1 INTRANET, NODE TO NODE	CONGESTION CONTROL, ROUTING
0 LINK CONTROL	ERROR HANDLING, LINK FLOW CONTROL

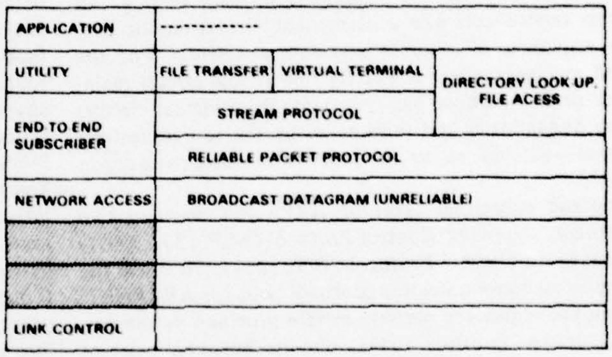


Fig. 2. ETHERNET protocol layering.

A. ETHERNET

In Fig. 2, we represent the Xerox ETHERNET protocol hierarchy. The basic link control mechanism is the ability of the interface device to detect conflict on a shared coaxial cable. If a transmitting interface detects that another interface is also transmitting, it immediately aborts the transmission. Hosts attached to the network interface present datagrams to be transmitted and are told if the datagram was aborted. Datagrams can be addressed to specific interfaces or to all of them. The end/end subscriber layer of protocol is split into two parts: a reliable datagram protocol in which each datagram is reliably delivered and separately acknowledged, and a stream protocol which can be thought of as a virtual circuit. This split is possible, in part, because there is a fairly large maximum datagram size (about 500 bytes) so that user applications can send datagrams without having to fragment and reassemble them. This makes the datagram service useful for many applications which might otherwise have to use the stream protocol. All higher level protocols, such as Virtual Terminal and File Transfer, are carried out in the hosts.

B. ARPANET

The ARPANET protocol hierarchy is shown in Fig. 3. The basic link control between packet switches treats the physical link as eight independent virtual links. This increases effective throughput, but does not necessarily preserve the order in which packets were originally introduced into the network. The intranet node-to-node protocols deal with adaptive routing decisions, store-and-forward service, and congestion control. Hosts have the option of either passing messages (up to

APPLICATION	RJE	ELECTRONIC MAIL	
UTILITY	TELNET	FTP	
END/END SUBSCRIBER	NCP	TCP	NVP/NVCP
NETWORK ACCESS	PERMANENT VIRTUAL CIRCUIT		DATAGRAM
INTRANET END/END	FLOW CONTROL, SEQUENCING, MESSAGE REASSEMBLY		▨
INTRANET NODE/NODE	ADAPTIVE ROUTING, STORE AND FORWARD, CONGESTION CONTROL		
LINK CONTROL	NON-SEQUENCED, MULTI-CHANNEL ERROR CONTROL		

Fig. 3. ARPANET protocol layering.

8063 bits of text) across the host/network interface, which will be delivered in sequence to the destination, or passing datagrams (up to 1008 bits of text) which are not necessarily delivered in sequence. The user's network access interface is datagram-like in the sense that no circuit setup exchange is needed even to activate the sequenced message service. In effect, this service acts like a permanent virtual circuit over which a sequence of discrete messages are sent. For the sequenced messages, there is exactly one virtual circuit maintained for each host/host pair. In fact, these virtual circuits are set up dynamically and terminated by the source/destination packet switches so as to improve resource utilization [38], [62].

The end/end subscriber layer of ARPANET contains two main protocols: Network Control Protocol (NCP, [39], [40]) and Transmission Control Protocol (TCP, [25]). NCP was the first interprocess communication protocol built for ARPANET. It relies on the sequenced message service provided by the network and derives multiple virtual circuits between pairs of hosts by multiplexing. The TCP can use either the sequenced message service or the datagram service. It does its own sequencing and end/end error control and derives multiple virtual circuits through extended addressing and multiplexing. TCP was designed for operation in a multinet environment in which the only service which reasonably could be expected was an unreliable, unsequenced datagram service.

To support experiments in packetized voice communication, two protocols were developed for use on the ARPANET. The Network Voice Protocol (NVP) and Network Voice Conferencing Protocol (NVCP) use the datagram service to achieve very low delay and interarrival time variance in support of digital, compressed packet speech (more on these protocols may be found in [41]). The NVP could be considered the basis for a generic protocol which could support a variety of real-time, end/end user applications.

The higher level utility protocols such as terminal/host protocol (TELNET, [40], [42]) and file transfer protocol (FTP, [40], [42]) use virtual circuits provided by NCP or TCP. The FTP requires one live interactive stream to control the data transfer, and a second for the data stream itself. Yet higher level applications such as electronic mail and remote job entry (RJE, [40], [42]) use mixtures of TELNET and FTP to effect the service desired. These protocols are usually put into the hosts. There is one anomaly, which occurs in many networks. Because terminal handling is required so frequently, a Terminal Interface Message Processor (TIP, [43]) was built. This device is physically integrated with the packet switch (IMP, [38]); it includes also the NCP and TELNET protocols.

END/END SUBSCRIBER	TERMINAL TO HOST
NETWORK ACCESS	VIRTUAL CIRCUIT
INTRANET END/END	▨
INTRANET NODE/NODE	FRAME DISASSEMBLY, REASSEMBLY, ROUTING, STORE/FORWARD, CONGESTION CONTROL
LINK CONTROL	FRAME-BASED ERROR CONTROL, RETRANSMISSION, SEQUENCING

Fig. 4. TYMNET protocol layering.

C. TYMNET

TYMNET (see Fig. 4) is one of the oldest of the networks in the collection described here [3]. Strictly speaking, it operates rather differently than other packet-switched networks, because the frames of data that move from switch to switch are disassembled and reassembled in each switch as an integral part of the store-and-forward operation. Nevertheless, the network benefits from the asynchronous sharing of the circuits between the switches in much the same way that more typical packet-switched networks do. The network was designed to support remote terminal access to time-shared computer resources. The basic service is the transmission of a stream of characters between the terminal and the serving host. A frame is made up of one or more blocks of characters, each block labeled with its source terminal identifier and length. The switch-to-switch layer of protocol disassembles each frame into its constituent blocks and uses a routing table to determine to which next switch the block should be sent. Blocks destined for the same next switch are batched together in a frame which is checksummed and sent via the link control procedure to the next switch. Batching the blocks reduces line overhead (the blocks share the frame checksum) at the expense of more CPU cycles in the switch for frame disassembly and reassembly.

The protocol between TYMNET switches also includes a flow control mechanism which, because of the fixed routes, can be used to apply back pressure all the way back to the traffic source. This is not precisely an end-to-end flow control mechanism, but a hop-by-hop back pressure strategy. Character blocks are kept in sequence along the fixed routes so that no resequencing is required as they exit from the network at their destinations. The network interface is basically a virtual circuit designed to transport character streams between a host and a terminal. The same virtual circuits can be used to transport character streams between hosts, which look to each other like a collection of terminals. Above the basic virtual circuit service, is a special echo-handling protocol which allows the host and the terminal handler in the "remote TYMSAT" to coordinate the echoing of the characters typed by a user.

D. PTT Networks

Many PTT networks, e.g., TELNET, TRANSPAC, DATAPAC, and EURONET use a particular network-access protocol, X.25 [28], [29] (see Fig. 5). This protocol has been recommended by the CCITT for public packet-switched data networks. X.25 is a three-part protocol consisting of a hardware electrical interface, X.21 [44], the digital equivalent of the usual V.24 or EIA-RS232C modem interface [45], a link control procedure, High Level Data Link Control (HDLC, [46]), and a packet-level protocol for effecting the setup, use, termination, flow, and error control of virtual circuits.

UTILITY	TERMINAL HANDLING X 28. X 29
END/END SUBSCRIBER	▨
NETWORK ACCESS	X 25. PERMANENT OR TEMPORARY VIRTUAL CIRCUITS
INTRANET. END END	MULTIPLE VIRTUAL CIRCUITS. FLOW CONTROL
INTRANET NODE NODE	ROUTING. STORE/FORWARD. CONGESTION CONTROL
LINK CONTROL	HDLC OR EQUIVALENT

Fig. 5. PTT protocol layering.

In all but the DATAPAC network, a fixed route for routing packets through the network is selected at the time the virtual circuit is created. "Permanent" virtual circuits are a customer option; if used, the setup phase is invoked only in the case of a network failure. Between source and destination packet switches, a virtual circuit protocol is operated which implements end-to-end flow control on multiple virtual circuits between pairs of packet switches. Up to 4096 virtual circuits between pairs of host ports can be maintained by each packet switch, as compared to the single virtual circuit provided by ARPANET (on which hosts can multiplex their own virtual circuits). This choice has a noticeable impact on the subscriber interface protocol which becomes complicated because the subscriber host and the packet switch to which it attaches must maintain a consistent view of the state of each virtual circuit in use.

To provide for echo control, user commands, code conversion, and other terminal-related services, these networks implement CCITT Recommendations X.28 [29] and X.29 [29] in a PAD (Packet Assembly and Disassembly unit). These protocols sit atop the virtual circuit X.25 protocol. In order to serve customers desiring a terminal-to-host service with character terminals, such as is provided by TYMNET or by the ARPANET (through the TIP), most of the PTT networks mentioned are developing a PAD unit. A matching X.29 (PAD control protocol) layer must be provided in hosts offering to service terminals connected to PAD's.

E. High Level Protocols

The X.25/X.28/X.29 protocol hierarchy does not include an end/end subscriber or high-level protocol layer. Some customers will, in fact, implement end-to-end protocols on top of the virtual circuit protocol, but others may not. Several attempts are being made to standardize protocols above the network access level. The ARPANET community has developed a Transmission Control Protocol [25] for internetwork operation to replace the Network Control Program (NCP) developed early in the ARPANET project. The International Federation of Information Processing (IFIP) has proposed a Transport Station through its Working Group 6.1 on Network Interconnection [47]; the proposal has been submitted to the International Standards Organisation (ISO) as a draft standard. In addition, other communities, e.g., the High Level Protocol Working Group in the UK, have devised protocols for Virtual Packet Terminals (VPT, [48]) and File Transport Protocol (FTP, [49]) which are intended to be network independent and which may be submitted to CCITT. The ISO study on "open systems architecture" and the proposed similar study by CCITT Study Group VII will attempt to evolve higher level protocol recommendations for existing and future data networks.

This brief summary of different network-protocol layerings is in no way comprehensive, but illustrates the diversity of protocol designs which can be found on nets providing different types of services to subscribers.

VI. Technical Interconnection Choices

A. The Issues

Beginning with the earliest papers dealing with strategies for packet-network interconnection [23]-[26], [32], the common objective of all the proposed methods is to provide the physical means to access the services of a host on one network to all subscribers (including hosts) of all the interconnected networks. Of course, limitations to this accessibility are envisaged, imposed either for administrative reasons or by the scarcity of resources. The achievement of this objective invariably requires that data produced at a source in one net be delivered and correctly interpreted at the destination(s) in another network. In an abstract sense, this boils down to providing interprocess communication across network boundaries. Even if a person is the ultimate source of the data, packet-switching networks must interpose some degree of software processing between the person and the destination service, even if only to assemble or disassemble packets produced by a computer terminal.

A fundamental aspect of interprocess communication is that no communication can take place without some agreed conventions. The communicating processes must share some physical transmission medium (wire, shared memory, radio spectrum, etc.), and they must use common conventions or agreed upon translation methods in order to successfully exchange and interpret the data they wish to communicate. One of the key elements in any network interconnection strategy is therefore how the required commonality is to be obtained. In some cases, it is enough to translate one protocol into another. In others, protocols can be held in common among the communicating parties.

In any real network interconnection, of course, a number of secondary objectives will affect the choice of interconnection strategy. For example achievable bandwidth, reliability, robustness (i.e., resistance to failures), security, flexibility, accountability, access control, resource allocation options, and the like can separately and jointly influence the choice of interconnection strategy. Combinations of strategies employing protocol standards and protocol translations at various levels of the layered protocol hierarchy are also likely possibilities.

There are a number of issues which must be resolved before a coherent network interconnection strategy can be defined. A list of some of these issues, which will be treated in more detail in succeeding sections, is:

1) level of interconnection;
2) naming, addressing, and routing;
3) flow and congestion control;
4) accounting;
5) access control;
6) internet services.

B. Gateways and Levels of Network Interconnection

The concept of a gateway is common to all network interconnection strategies. The fundamental role of the gateway is to terminate the internal protocols of each network to which it is attached while, at the same time, providing a common

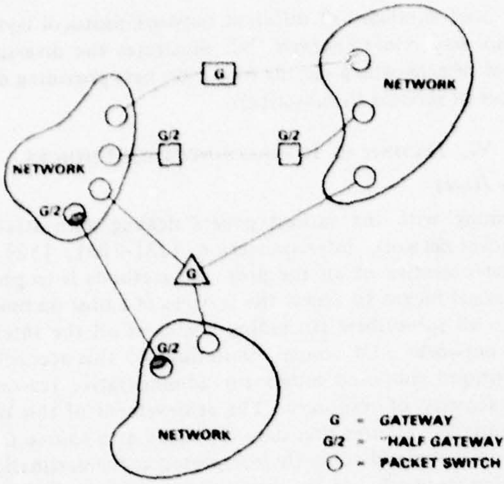


Fig. 6. Various gateway configurations.

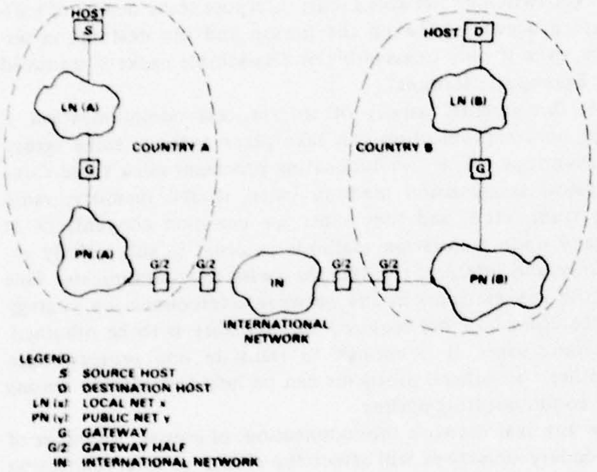


Fig. 7. International packet-networking model.

ground across which data from one network can pass into another. However, the choice of functions to be performed in the gateway varies considerably among different interconnection strategies (see Fig. 6). The term "gateway" need not imply a monolithic device which joins a pair of networks. Indeed, the gateway may merely be software in a pair of packet switches in different networks, or it may be made up of two parts, one in each network (a sort of "gateway half"). In the latter case, the two halves might be devices separate and distinct from the network packet switches or might be integrated with them. Furthermore, a gateway might interconnect more than two networks. In the material which follows, every attempt has been made to avoid any implicit choice of gateway implementation. It is worth pointing out, however, that the "half gateway" concept is highly attractive from both a technical and a purely administrative point of view. Technically, each half could terminate certain levels of protocol of the net to which it is attached. Administratively each half could be the responsibility of the network to which it belongs. Then the only matters for jurisdictional negotiation are the physical medium by which the half-gateways exchange data, and the format and protocol of the exchange.

It is important to realize that typical applications may involve three or more networks. Where local networks are used, they will usually need to be interconnected to realize the benefits of interorganizational data exchange. In most countries, such interconnections will only be permitted through a public network. Thus for a typical national situation, three networks and two gateways will be involved in providing the desired host-to-host communication.

The international picture is similar, except that more networks are likely to be involved. Shown in Fig. 7, the path from a host, S, on local network $LN(A)$ in country A, passes through a public network, $PN(A)$ in country A, through an international network IN, through a public network $PN(B)$ in country B, and finally through a local network, $LN(B)$, to the destination host, D. There are four internetwork gateways involved. It is this model involving multiple gateways that guides us away from network interconnection methods which rely on the source and destination hosts being in adjacent networks connected by the mediation of a single gateway.

1) Common Subnet Technology (Packet Level Interconnection): The level at which networks are interconnected can be determined by the protocol layers terminated by the gateway. For example, if a pair of identical networks were to be interconnected at the interpacket-switch level of protocol, we might illustrate the gateway placement as shown in Fig. 8. Here the "gateway" may consist only of software routines in the adjacent packet switches, e.g., $P(A)$ and $P(B)$, which provide accounting, and possibly readdressing functions. The contour model of protocol layer is useful here since it shows which levels are common to the two networks and which levels could be different. In essence, those layers which are terminated by the gateways could be different in each net, while those which are passed transparently through the gateway are assumed to be common in both networks. This network interconnection strategy requires that the internal address structure of all the interconnected networks be common. If, for example, addresses were composed of a network identifier, concatenated with a packet-switch identifier and a host identifier, then addressing of objects in each of the networks would be straightforward and routing could be performed on a regional basis with the network identifiers acting as the regional identifiers, if desired. Alternatively, two identical networks could adopt a common network name and assign nonduplicative addresses to each of the packet switches in both networks. This may require that addresses in one network be changed.

The strategy described above might be called the "common subnetwork strategy," since, in the end, subscribers of the newly formed joint network would essentially see a single network. This strategy does not rule out the provision of special access control mechanisms in the gateway nodes which could filter traffic flowing from one network into the other. Similarly, the gateway nodes could perform special internetwork traffic accounting which might not normally be performed in a subnet switching node. This network interconnection method is limited to those cases in which the nets to be connected are virtually identical, since the gateways must participate directly in all the subnet protocols. The end-to-end subnet protocols (e.g. source/destination packet-switch protocols) must pass transparently through the gateways to permit interactions between a source packet switch in one net and a destination packet switch in another. The resulting network presents the same network access interface to all

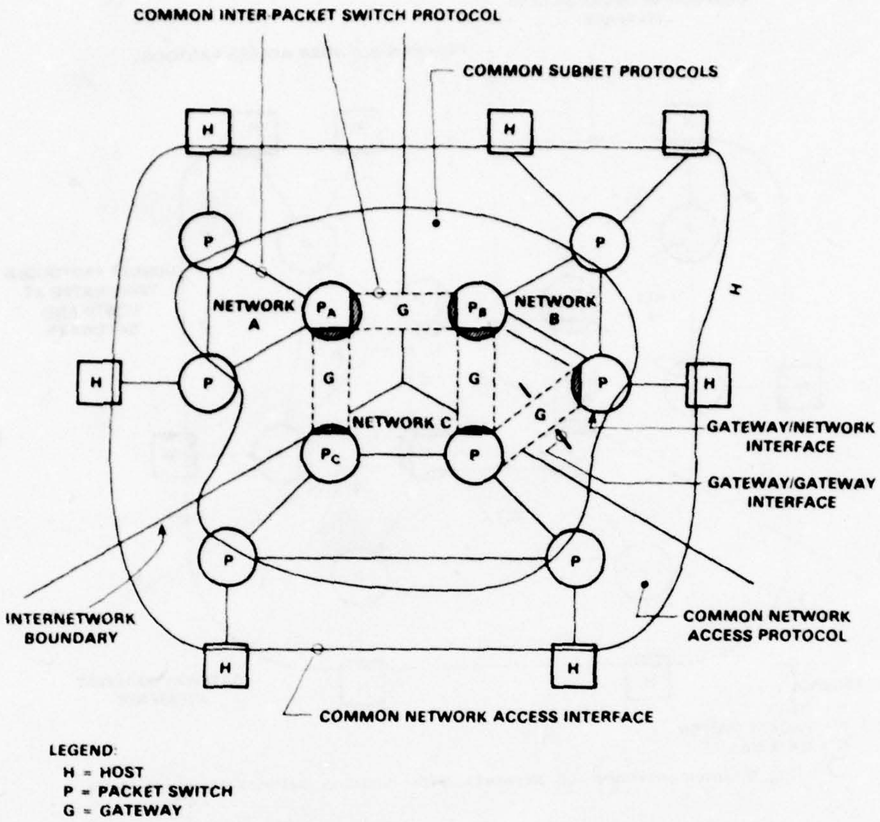

COMMON INTER-PACKET SWITCH PROTOCOL
COMMON SUBNET PROTOCOLS
H
P
NETWORK A
P_A
G
P_B
NETWORK B
NETWORK C
P_C
GATEWAY/NETWORK INTERFACE
GATEWAY/GATEWAY INTERFACE
INTERNETWORK BOUNDARY
COMMON NETWORK ACCESS PROTOCOL
COMMON NETWORK ACCESS INTERFACE
LEGEND:
H = HOST
P = PACKET SWITCH
G = GATEWAY

Fig. 8. Interconnection of common subnetworks.

subscribers, and this leads us to the next example which is based on the concept of a common network access interface.

2) Common Network Access Interfaces: If the subnetwork protocols are not identical, the next opportunity to establish internetwork commonality is at the network access interface. This is illustrated in Fig. 9. Each network is assumed to have its own intranet protocols. However, each network presents the same external interface to subscribers. This is illustrated by showing a common interface passing through all hosts, marked "common network access interface" in the figure.

Once again, the gateway could be thought of as software in adjacent packet switches. Each gateway is composed of two halves formed by linking the packet switches of two nets together. However, in this case, the subnetwork protocols are terminated at the gateway so that the intergateway exchange looks more like network access interaction than a node-to-node exchange. This is the approach taken by CCITT with its X.25 packet network interface recommendation and X.75 intergateway exchange recommendation.

It is important to note that the intergateway interface could be similar to the standard network access interface, but it need not necessarily be identical.

There are two basic types of network interface currently in use: 1) the datagram interface [31]; and 2) the virtual circuit interface [32]. The details of these generic interface types vary in different networks; some networks even offer both types of interface. In some, the interface to use may be chosen at subscription time; in others it may be possible for a subscriber to select the access method dynamically.

A datagram interface allows the subscriber to enter packets into the network independent of any other packets which have been or will be entered. Each packet is handled separately by the network. A virtual circuit interface requires an exchange of control information between the subscriber and the network for the purpose, for example, of setting up address translation tables, setting up routes or preallocating resources, before any data packets are carried to the destination. Some networks may implement a *fast select* virtual circuit interface in which a circuit setup request is sent together with the first (and possibly last) data packet. Other control exchanges would be used to close the resulting virtual circuits set up in this fashion.

It is essential to distinguish datagram and virtual circuit services from datagram and virtual circuit interfaces. A datagram service is one in which each packet is accepted and treated by the network independently of all others. Sequenced delivery is not guaranteed. Indeed, it may not be guaranteed that all datagrams will be delivered. Packets may be routed independently over alternate network paths. Duplicate copies of datagrams might be delivered.

Virtual circuit service tries to guarantee the sequenced delivery of the packets associated with the same virtual circuit. It typically provides to the host advice from the network on flow control per virtual circuit as opposed to the packet-by-packet acceptance or rejection typical of a datagram service. If the network operation might produce duplicate packets, these are filtered by the destination packet switch before delivery to the subscriber. Duplicate packet creation is a

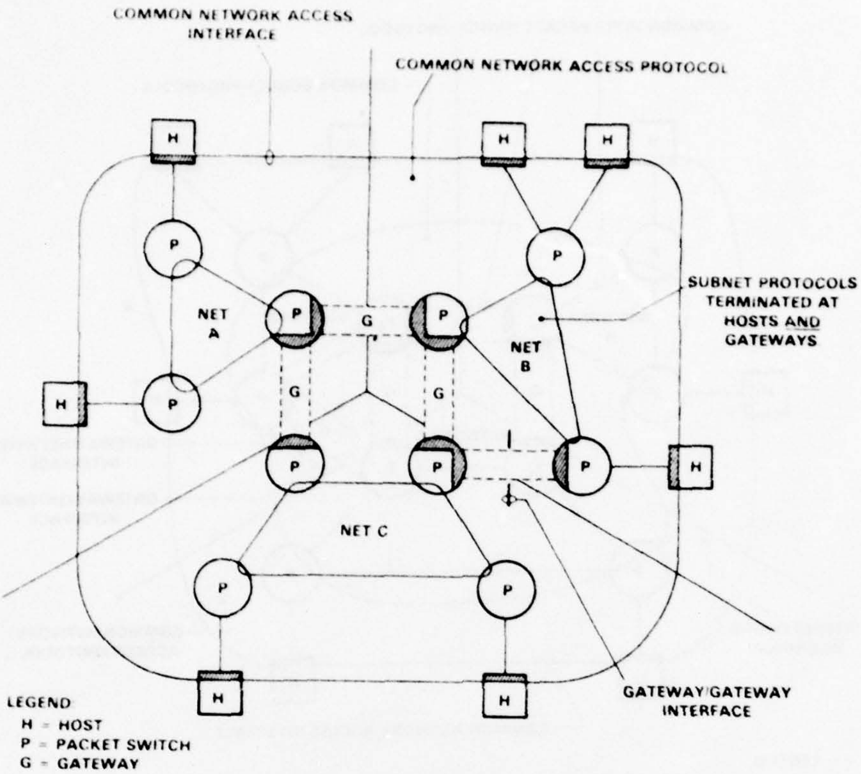

COMMON NETWORK ACCESS INTERFACE
COMMON NETWORK ACCESS PROTOCOL
H
P
G
NET A
NET B
NET C
SUBNET PROTOCOLS TERMINATED AT HOSTS AND GATEWAYS
GATEWAY/GATEWAY INTERFACE
LEGEND
H = HOST
P = PACKET SWITCH
G = GATEWAY

Fig. 9. Interconnection of networks with common network-access interfaces.

common phenomenon as in packet-switched store-and-forward systems. The basic mode of operation is to forward a packet to the next switch and await an acknowledgment. After a timeout, the packet is retransmitted. If an acknowledgment is lost due to line noise, for example, then two copies of the packet would have been transmitted. Even if the next switch is prepared to filter duplicates out, a network which uses adaptive routing can deliver a duplicate packet to the periphery of the network. For example, if a packet switch receives a packet successfully but the line to the sender breaks before the receiver can acknowledge, the sender may send another copy to a *different* packet switch. Both packet copies may be routed and delivered to the destination packet switch where final duplicate filtering would be needed if virtual circuit service is being provided.

Some networks offer both a datagram and a virtual circuit service; some offer a single interface, but different services. For example, the ARPANET has a basic datagram interface. However, the subnetwork will automatically provide a sequenced virtual circuit service (i.e., packets are kept in sequence when they are delivered to the destination) if the packet is marked appropriately. Otherwise, packets are not delivered in sequence nor are packet duplicates or losses, except for line by link correction, recovered within the network for nonsequenced types of traffic.

By contrast, TRANSPAC offers a virtual circuit interface and service. Subscribers transmit "call request" packets containing the full destination address to the packet switch. The request packet is forwarded to the destination, leaving behind a fixed route. The destination subscriber returns a "call accepted" packet which is delivered to the caller. As a result of this exchange, the source subscriber has associated a "logical channel number" or LCN, with the full source-destination addresses. Thus subsequent packets to be sent on the same logical channel are identified by the LCN and are kept in sequence when delivered to the destination.

Finally, it is possible to implement a datagram-like service using a virtual circuit interface. In this case, the exchange of *request* and *accept* packets might be terminated at the subscriber's local packet switch, so that even if packets were not delivered in sequence they might employ abbreviated addressing for local subscriber and packet-switch interaction.

If network interaction is to be based on a standard interface, then agreement must be reached both on the interface and an associated service or services. Furthermore, a common addressing system is needed so that a subscriber on one network can address a packet to a subscriber on any other network. A weaker assumption could be made but we are deliberately assuming a truly common service, interface, and addressing mechanism. We will return to this topic in a later section.

The choice of a standard network service through which to effect network interconnection has a primary impact on the flexibility of implementable network interconnection methods. We will consider two choices: datagram service and virtual circuit service.

a) Datagram service as a standard for network interconnection: For this case, it is assumed that every network offers a common datagram service. A uniform address space makes it possible for subscribers on any network to send packets addressed to any other subscriber on a connected network. Packets are routed between subscriber and gateway and between gateways based on the destination address. No attempt is

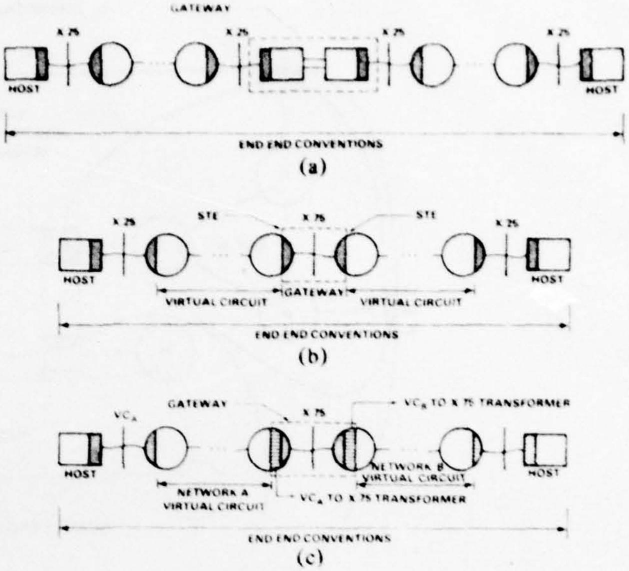


Fig. 10. Virtual circuit network interconnection strategies. (a) Subscriber-based gateway. Internet source and destination carried in user data field of X.25 call set-up packets. (b) X.75 based gateway. Note how much of the X.25 VC service is terminated at the STE. (c) X.75-based gateways with general virtual circuit networks.

made to keep the datagrams in any order in transit or upon delivery to the destination. Individual datagrams may be freely routed through different gateways to recover from failures or to allow load-splitting among parallel gateways joining a pair of networks.

The gateway/gateway interface may be different than the network access interface, if need be (see Fig. 9).

This strategy requires that all networks implement a common interface for subscribers. The simplicity and flexibility of the datagram interface strategy is offset somewhat by the need for all networks to implement the same interface. This is true for the pure virtual circuit interface strategy as well, as will be shown below.

One of the problems which has to be faced with any network interconnection strategy is congestion control at the gateways. If a gateway finds that it is unable to forward a datagram into the next network, it must have a way of rejecting it and quenching the flow of traffic entering the gateway en route into the next network. The quenching would typically take the form of an error or flow control signal passing from one gateway half to another on behalf of the associated network. Similar signals could be passed between subscribers and the packet network for similar reasons. Since datagram service does not undertake to guarantee end/end reliability, it is possible to relieve momentary congestion by discarding datagrams, as a last resort.

b) Virtual circuits for network interconnection: Another alternative standard network service which could be used for network interconnection is virtual circuit service (Fig. 10). Independent of the precise interface used to "set up" the virtual circuit, a number of implementation issues immediately arise if such a service is used as a basis for network interconnection.

Since it is intended that all packets on a virtual circuit be delivered to the destination subscriber in the same sequence as they were entered by the source subscriber, it is necessary that either: 1) all packets belonging to the same virtual circuit take the same path from source subscriber, through one or more gateways, to destination subscriber; or 2) all packets contain sequence numbers which are preserved end-to-end between the source DCE in the originating network and the destination DCE in the terminating network.

In the first case, virtual circuits are set up and anchored to specific gateways so that the sequencing of the virtual circuit service of each network can be used to preserve the packet sequence on delivery. This results in the concatenation of a series of virtual circuits through each gateway and, therefore, the knowledge of each virtual circuit at each gateway (since the next gateway to route the packet through must be fixed for each virtual circuit).

In the second case, there is no need to restrict the choice of gateway routing for each virtual circuit since the destination DCE will have sufficient information to resequence incoming packets prior to delivery to the destination subscriber.

In either case, the destination DCE will have to buffer and resequence packets arriving out of order due either to disordering within the last network or to alternate routing among networks, if this is permitted. Some networks may keep packets in sequence as they transit the network. This will only be advantageous at the destination DCE if the packets enter the network in the desired sequence. If such a service is relied upon in the internet environment, then each gateway must assure that on entry to such a net, the packets are in the desired order for delivery to a destination subscriber or another gateway.

The buffering and resequencing of packets within the networks or at gateways introduces substantial variation in buffer space requirements, packet transit delays, and the potential for buffer lockups to occur [50], [51], [61].

If packets for a specific virtual circuit are restricted to pass through a fixed series of gateways, and if a standard flow-control method is agreed upon as part of the virtual circuit service, then it is possible for each internet gateway to participate in end-to-end flow control by modifying the flow control information carried in packets carried end-to-end from the source DCE to the destination DCE. Consequently, a gateway may be able to adjust the amount of traffic passing through it and thereby achieve a kind of internet gateway congestion control. If this is done by allocating buffer space for "outstanding" packets, then either the gateways must guarantee the advertised buffer space or there must be a retransmission capability built into the internet virtual circuit implementation, perhaps between source DCE and destination DCE or between DCE's and gateways.

Such a mechanism does not, however, solve the problem of network congestion unless the gateway-flow control decisions take into account resources both in the gateway and in the rest of the network. Although it is tempting to assume that virtual circuit-flow control can achieve internetwork congestion control, this is by no means clear, and is still the subject of considerable research.

As a general rule, compared to the datagram method, the virtual circuit approach requires more state information in each gateway, since knowledge of each virtual circuit must be maintained along with flow control and routing information. The usual virtual circuit interface is somewhat more complex for subscribers to implement as well, because of the amount of state information which must be shared by the subscriber and the local DCE. For example, implementations of the X.25

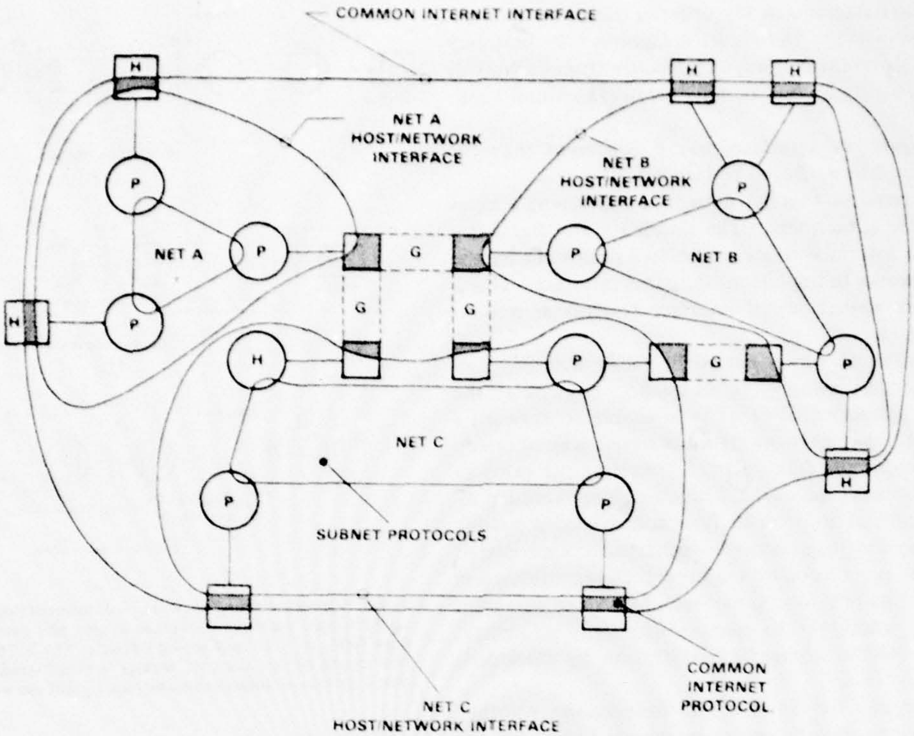

COMMON INTERNET INTERFACE
NET A HOST/NETWORK INTERFACE
NET B HOST/NETWORK INTERFACE
H
P
G
NET A
NET B
NET C
SUBNET PROTOCOLS
COMMON INTERNET PROTOCOL
NET C HOST/NETWORK INTERFACE

Fig. 11. Common internet interface.

interface protocol have been privately reported by Computer Corporation of America and University College London to require 4000–8000 words of memory on Digital Equipment Corporation PDP-11 computers. By contrast, the ARPANET and Packet Radio Network datagram interfaces require 500–1000 words of memory on the same machine. For internetwork operation, this may be even more burdensome, since any failure at a gateway may require a subscriber-level recovery through an end-to-end protocol, in addition to the virtual circuit interface software, as is shown in [52].

Nevertheless, it may be advantageous to consider internetworking standards which usefully employ both datagram and virtual circuit interfaces and services. For example, some special internet services such as multidestination delivery may be more efficient if they are first set up by control exchanges between the subscriber and the local network and perhaps gateways as well. Once set up, however, a datagram mode of operation may be far more efficient than maintaining virtual circuits for all destinations. Implicit virtual circuits which are activated by simple datagram-like interfaces are also attractive for very simple kinds of terminal equipment.

If it is not possible for all networks to implement a common network-access interface, then the next opportunity is to standardize only the objects which pass from one net to the next and to minimize any requirements for the sequencing of these objects as they move from net to net.

3) General Host Gateways: In this model, a gateway is indistinguishable from any other network host and will implement whatever host/network interface is required by the networks to which it is attached. For many networks, this may be X.25, but the strategy does not rely on this. The principle assumption is that packet networks are at least capable of carrying subscriber packets up to some maximum length, which may vary from network to network. It is specifically not assumed that these packets will be delivered in order through intermediate networks and gateways to the destination host. This minimal type of service is often termed "datagram" service to distinguish it from sequenced virtual circuit service. A detailed discussion of the tradeoff between datagram and virtual circuit types of networks is given elsewhere [52].

The basic model of network interconnection for the datagram host gateway is that internetwork datagrams will be carried to and from hosts and gateways and between gateways by encapsulation of the datagrams in local network packets. Pouzin describes this process generically as "wrapping" [37]. The basic internetwork service is therefore a datagram service rather than a virtual circuit service. The concept is illustrated in Fig. 11.

Datagram service does not offer the subscriber as many facilities as virtual circuit service. For example, not all datagrams are guaranteed to be delivered, nor do those that are delivered have to be delivered in the sequence they were sent. Virtual circuits, on the other hand, do attempt to deliver all packets entered by the source in sequence to the destination. These relaxations allow dynamic routing of datagrams among multiple, internetwork gateways without the need for subscriber intervention or alert.

The internet datagram concept gives subscribers access to a basic internet datagram service while allowing them to build more elaborate end-to-end protocols on top of it. Fig. 12 illustrates a possible protocol hierarchy which could be based on the internet datagram concept. The basic internet datagram service could be used to support transaction protocols or real-time protocols (RTP) such as packet-voice protocols (PVP) which do not require guaranteed or sequenced data

UTILITY	FTP	VTP	RTP	VP
END/END SUBSCRIBER	END/END VIRTUAL CIRCUIT		END/END DATAGRAM	
INTERNET ACCESS	INTERNET DATAGRAM			
NETWORK ACCESS	NETWORK SPECIFIC			
INTRANET, END-END	NETWORK SPECIFIC			
INTRANET, NODE-NODE	NETWORK SPECIFIC			
LINK CONTROL	NETWORK SPECIFIC			

Fig. 12. Protocol layering with internetwork datagrams.

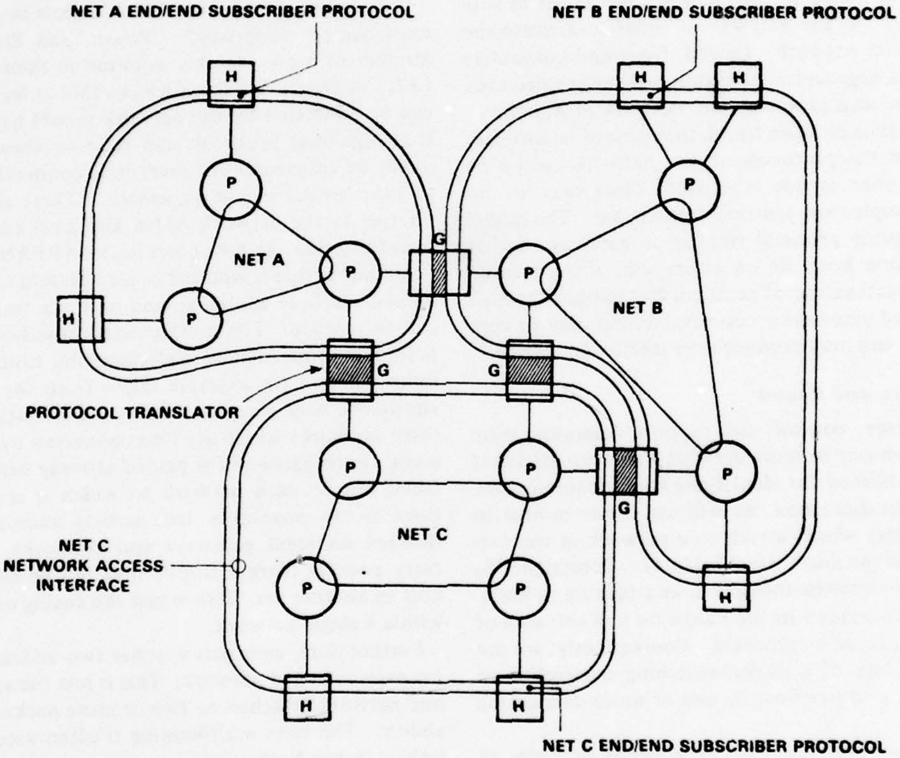

NET A END/END SUBSCRIBER PROTOCOL
NET B END/END SUBSCRIBER PROTOCOL
H
H
H
P
P
P
P
G
NET A
P
NET B
H
G
G
PROTOCOL TRANSLATOR
P
P
P
G
NET C
NET C
NETWORK ACCESS
INTERFACE
P
P
H
H
H
NET C END/END SUBSCRIBER PROTOCOL

Fig. 13. Host protocol translation gateway.

delivery; reliable, sequenced protocols could be constructed above the basic internet datagram service to perform end/end sequencing and error handling. Applications such as virtual terminal protocols (VTP) [40], [42], [48] or file-transfer protocols [40], [42], [49] could be built above a reliable, point-to-point, end/end service which is itself built atop internet datagrams. Under this strategy, the basic gateway functions are the encapsulation and decapsulation of datagrams, mapping of internet source/destination addresses into local network addresses and datagram routing. Gateways need not have any knowledge of higher level protocols if it is assumed that protocols above the internet datagram layer are held in common by the communicating hosts. Datagrams can be routed freely among gateways and can be delivered out of sequence to the destination host.

The basic advantage of this strategy is that almost any sort of network can participate, whether its internal operation is datagram or virtual circuit oriented. Furthermore, the strategy offers an easy way for new networks to be made "backwards compatible," with older ones while allowing the new ones to employ new internal operations which are innovative or more efficient.

Every subscriber must implement the internet datagram concept for this strategy to work, of course. The same problem arises with the standard network interface strategy since all subscribers must implement the same network interface.

4) Protocol Translation Gateways: It would be misleading to claim that the concept of protocol translation has not played a role in the discussion thus far. In a sense, the encapsulation of internet datagrams in the packet format of each intermediate network is a form of protocol translation. The basic packet carrying service of one network is being translated into the next network's packet carrying service (see Fig. 13). This concept could be extended further. For example, if two networks have a virtual circuit concept, one implemented within the subnetwork and the other through common host/host protocols, it might be possible, at the gateway between the nets, to map one network's virtual circuit into the other's. This same idea could be applied to higher level protocol mappings as well; for instance, the virtual terminal protocol for one network might be transformed into that of another "on the fly."

The success of such a translation strategy depends in large part on the commonality of concept between the protocols to be translated. Mismatches in concept may require that the service obtained in the concatenated case be a subset of the services obtainable from either of the two services being translated. Extending such translations through several gateways can be difficult, particularly if the protocols being translated do not share a common address space for internetwork sources/destinations. In the extreme, this strategy can result in subscribers "logging in" to the gateway in order to activate the protocols of the next network. Indeed, front-end computers could be considered degenerate translation gateways since they transform host/front-end protocols into network protocols.

There are circumstances when translation cannot be avoided. For instance, when the protocols of one network cannot be modified, but internet service is desired, there may be no alternative but to implement protocol translations. The model typically used to guide protocol translation gateways is that the source/destination hosts lie on either side of the translation gateway. Concatenation of protocol translations through several networks and gateways is conceivable, but may be very difficult in practice and may produce very inefficient service.

C. Names, Addresses, and Routes

In order to manage, control, and support communication among computers on one or more networks, it is essential that conventions be established for identifying the communicators. For purposes of this discussion, we will use the term *host* to refer to all computers which attach to a network at the network-access level of protocol (see Table I). Subscribers to terminal-access services can be thought of as attaching to hosts, even if the host is embedded in the hardware and software of a packet switch as a layer of protocol. Consequently, we can say that the basic task of a packet-switching network is to transport data from a source host to one or more destination hosts.

To accomplish this task, each network needs to know to which destination packets are to be delivered. Even in broadcast nets such as the ETHERNET, this information is necessary so that the destination host can discriminate packets destined for itself from all others heard on the net. At the lowest-protocol levels it is typical to associate destinations with *addresses*. An address may be simply an integer or it may have more internal structure.

At higher levels of protocol, however, it is more common to find text strings such as "MULTICS" or "BBN-TENEX" used as *names* of destinations. Application software, such as electronic mail services, might employ such names along with more refined destination identifiers. For example, one of the authors has an electronic mailbox named "KIRSTEIN at ISI" located in a computer at the University of Southern California's Information Sciences Institute.

Typically, application programs transform names into addresses which can be understood by the packet-switching network. The networks must transform these addresses into *routes* to guide the packets to their destination. Some networks bind addresses to routes in a relatively rigid way (e.g., setting up virtual circuits with fixed routing) while others determine routes as the packets move from switch to switch, choosing alternate routes to bypass failed or congested areas of the network. Broadcast networks need not create routes at all (e.g., SATNET).

In simple terms, a *name* tells what an object is; an *address* tells where it is; and a *route* tells how to get there [54]. A simple model involving these three concepts is that hosts transform names into addresses and networks transform addresses into routes (if necessary). However, this basic model does leave a large number of loose ends. The subject is so filled with issues that it is not possible in this paper to explore them all in depth. In what follows, some of the major issues are raised and some partial resolutions are offered.

One major question is "Which objects in the network should have names? addresses?" Pouzin and Zimmermann offer a number of views on this question in their paper in this issue [37]. A generic answer might be that at least all objects which can be addressed by the network should have names as well so that high-level protocols can refer to them. For example, it might be reasonable for every host connection on the network to have an name and an address. There also may be objects internal to the network which also have addresses such as the statistics-gathering *fake hosts* in the ARPANET [38].

A related issue is whether objects should or can have multiple names, multiple addresses, and multiple routes by which they can be reached. The most general resolution of this issue is to permit multiple names, addresses, and routes to exist for the same object. An example taken from the multinetwork environment may serve to illustrate this notion. Fig. 6 shows three networks which are interconnected by a number of gateways. Each gateway (or pair of gateway halves) has two interfaces, one to each network to which it is attached. Plainly there is the possibility that several alternate routes passing through different gateways and networks could be used to carry packets from a source host in one net to a destination host in another net. This is just the analog of alternate routing within a single network.

Furthermore, each gateway has two addresses, typically one for each attached network. This is just the analog of a host on one network attached to two or more packet switches for reliability. The term *multihoming* is often used to refer to multiply attached hosts.

Finally, it may be useful to permit a gateway to have more than one name, for example, one for each network to which it is attached. This might allow high-level protocols to force packets to be routed in certain ways for diagnostic or other reasons. Multiple naming also allows the use of nicknames for user convenience. Many of these same comments would apply to hosts attached to multiple networks.

An interesting addressing and routing problem arises in mobile packet radio networks. Since hosts are free to move about, the network will need to dynamically change the routes used to reach each host. For robustness, it is also desirable that hosts be able to attach dynamically to different packet radios. Thus failure of a packet radio need not prevent hosts from accessing the network. This requires that host names and perhaps host addresses be decoupled from packet radio addresses. The network must be able to search for hosts or alternatively, hosts must "report-in" to the network so that their addresses can be associated with the attached packet radio to facilitate route selection based on host address. This is just a way of supporting *logical host addressing* rather than using the more common *physical host addressing* in which a host's address is an extension of the packet-switch address.

A crucial issue in network interconnection is the extent to which it should or must impact addressing procedures which are idiosyncratic to a particular network. It is advantageous not to require the subscribers on each network to have detailed knowledge of the network address *structure* of all interconnected networks. One possibility is to standardize an internetwork address structure which can be mapped into local network addresses as needed, either by subscribers, by gateways or by both. Subscribers would know how to map internetwork service names into addresses of the form NETWORK/SERVER. Subscribers need not know the fine structure of the SERVER field. Gateways would route packets on the basis of the NETWORK part of the address until reaching a gateway attached to the network identified by NETWORK. At this point, the gateway might interpret the SERVER part of the address, as necessary, to cause the packet to be delivered to the desired host.

The addressing strategy presently under consideration by CCITT (X.121, [30]) is based on the telephone network. Up to 14 digits can be used in an address. The first 4 digits are a "destination network identification code" or DNIC. Some countries are allocated more than one DNIC (the United States has 200). The remaining ten digits may be used to implement a hierarchical addressing structure, much like the one used in the existing telephone network.

Since the CCITT agreements are for international operation, it might be fair to assume that the United States will not need more than 200 public network identifiers. However, this scheme does not take into account the need for addressing private networks. The private networks, under this addressing procedure will most likely appear to be a collection of one or more terminals or host computers on one or more public networks. It is too early to tell how much this asymmetry in addressing between public and private networks will affect private multinetwork protocols.

A related problem which is not unique to network interconnection has to do with addressing (really multiplexing and demultiplexing) at higher protocol levels. The public carriers tend to offer services for terminal as well as host access to network facilities. This typically means that addresses must be assigned to terminals. The issue is whether the terminal address should be associated with or independent of the protocols used to support terminal-to-host communication.

The present numbering scheme would not distinguish between a host address and a terminal address. A host might have many addresses, each corresponding to a process waiting to service calling terminals.

There has been discussion within CCITT concerning "subaddressing" through the use of a user data field carried in virtual call "setup" packets. This notion would support the concept of a single host address with terminal or process level demultiplexing achieved through the use of the user data field subaddressing.

It seems reasonable to predict that, as terminals increase in complexity and capability, it will eventually be attractive to support multiple concurrent associations between the terminal and several remote service facilities. Applications requiring this capability will need terminal multiplexing conventions beyond those currently provided for in the CCITT recommendations.

To simplify implementations of internet protocol software, it is essential to place bounds on the maximum size of the NETWORK/SERVER address. Otherwise, subscribers may have to construct name-to-address mapping tables with arbitrarily large and complex entries.

Even if all these issues are resolved, there is still a question of "source routing" in which a subscriber defines the route to be taken by a particular packet or virtual circuit. Depending on the range of internetwork services available, a subscriber may want to control packet routes. It is not yet clear how such a capability will interact with access control conventions, but this may be a desirable capability if gateways are not able to automatically select routes which match user service requirements.

D. Flow and Congestion Control

For purposes of discussion, we distinguish between flow and congestion control. Flow control is a procedure through which a pair of communicators regulate traffic flowing from source to destination (each direction possibly being dealt with separately). Congestion control is a procedure whereby distributed network resources, such as channel bandwidth, buffer capacity, CPU capacity, and the like are protected from oversubscription by all sources of network traffic. In general, the successful operation of flow-control procedures for every pair of network communicants does not guarantee that the network resources will remain uncongested.

In a single network, the control of flow and congestion is a complex and not well understood problem. In a multinetwork environment it is even more complex, owing to the possible variations in flow and congestion control policies found in each constituent network. For example, some networks may rigidly control the input of packets into the network and explicitly rule out dropping packets as a means of congestion control. At the other extreme, some networks may drop packets as the sole means of congestion control.

At this stage of development, very little is known about the behavior of congestion in multiply interconnected networks. It is clear that some mechanisms will be required which permit gateways and networks to assert control over traffic influx especially when a gateway connects networks of widely varying capacity. This problem is likely to be most visible at gateways joining high speed local networks to long-haul public nets. The peak rates of the local nets might exceed that of the long-haul nets by factors of 30-100 or more. Generic procedures are needed for gateway/network and gateway/gateway flow and congestion control. Such problems also show up in single networks, but are amplified in the multinetwork case.

E. Accounting

Accounting for internetwork traffic is an important problem. The public networks need mechanisms for revenue sharing and subscribers need simple procedures for verifying the accuracy of network-provided accountings.

The public packet-switching networks appear to be converging on procedures which account for subscriber use on the basis of the number of virtual circuits created during the accounting period and the number of packets sent on each virtual circuit. Indeed, it has been argued that accounting on the basis of virtual circuits at gateways requires less overhead than accounting on a pure datagram basis [32]. Scenarios can be cited which support the opposite conclusion.

Suppose there is a choice between setting up virtual circuits for each transaction and sending a datagram for each transaction, and that virtual circuit accounting includes information on each virtual circuit setup (as in the present telephone network). If datagram accounting simply accumulates the number of datagrams sent between particular sources and destinations without regard to the time at which they are sent, then the amount of accounting information which is collected for the datagram case will be substantially less than for the virtual circuit case. In the limit (i.e., one packet per transaction), the virtual circuit accounting information is proportional to $2N$, where N is the number of transactions, while for the datagram case, it is proportional to log N (base 2). This is simply because the datagram case only sums counts for traffic between source/destination pairs while the virtual circuit accounting would identify start/stop times for each virtual circuit.

Alternatively, if the bulk of the traffic involves a large number of packets per transaction, then the two accounting procedures would accumulate more nearly the same information since each would predominantly involve accounting for packet flow.

If it is chosen not to account for virtual circuit duration, but merely to account independently for the number of virtual circuits and the number of packets sent between source/destination pairs, then the virtual circuit accounting would be closer to the datagram case.

The important conclusion to be drawn is that accounting for datagrams is generally less complex than accounting for virtual circuits, but that the two can be made arbitrarily similar by suitable choice of the details of the accounting information collected.

F. Access Control

In multinetwork environments, it may be necessary for each network to establish and enforce a policy for "out-of-network" routing. For example, a public network might conclude agreements with other networks regarding the type and quantity of traffic it will forward into other networks. This might even be a function of the time-of-day. Consequently, mechanisms are needed which will permit networks to prevent traffic from entering or leaving or to meter the type and rate of traffic passing into or out of the network.

Another example of the need for control arises with the possibility of third-party routing. That is, traffic destined from network A to network B is routed through network C. It cannot be assumed that all networks have gateways to all others. However, some nets may want to limit the amount of *transit* traffic they carry. There may be explicit agreements among a subset of the nets regarding revenue sharing for transit services. If a particular network does not have a revenue-sharing agreement with the particular source/destination networks of a given virtual circuit or datagram, then it must be able to reject the offending traffic if it so chooses.

There does not seem to be any technical barrier to separating the access control policy decision mechanism from the enforcement of the policy. For example, a gateway might simply enforce policy by sending traffic for which it has no known access rules to an *access controller*. If we adhere to the model that gateways have two *halves*, then each half deals with the network to which it is connected. The access controller can either dynamically enable the flow by causing table entries at the gateways which permit the flow to be created or it can tell the gateway to reject all further traffic of that type.

Clearly, access control policies will affect routing strategies, so this adds a complicating factor into any internetwork routing strategy implemented by the gateways. At present, very little experience has been accumulated with internet access control and routing policies. For the most part, agreements among public networks have been bilateral and transit routing has been treated as a very special case. When EURONET [6] becomes operational, this problem will be particularly important to solve.

G. Internet Services

It is by no means clear what set of services should be standardized and available from, at least, all public data networks. The current CCITT recommendations provide for virtual circuit service and terminal access service on all public packet-switching networks.

Although the recommendations (X.3, X.25) provide for *fragmentation* of packets being delivered to a subscriber on a virtual circuit, the current X.75 gateway draft recommendation uses an agreed maximum packet size of 128 octets of data, not including the header. This agreement avoids for the moment the need to fragment packets crossing a network boundary, as long as all subscribers recognize that the maximum length internetwork packet allowed is 128 octets. Bilateral exceptions to this rule may develop but neither a fixed size nor a collection of special cases represent a very general solution to this problem.

It has been argued [25] that a general scheme for dealing with fragmentation is desirable so that new network technologies supporting larger packet sizes can be easily integrated into the multinetwork environment.

Apart from fragmentation, there are a set of special services such as multidestination addressing and broadcasting which could be used to good advantage to support multinetwork applications such as teleconferencing, electronic mail distribution, distributed file systems, and real-time data collection. Other services such as low delay, high reliability, high bandwidth, and high priority are also candidates for standardization at the internet level.

As in the case of access control, selection of such services might constrain the choice of packet routing to networks capable of supporting the desired services. Once again, very little experience with standard internet services has been accumulated so this subject is still a topic for research. For the most part, terminal-to-host services have been successfully offered across network boundaries using nearly all of the network interconnection methods described in this paper. It remains to be seen whether more complex applications can be equally well supported.

VII. X.25/X.75–The CCITT Strategy for Network Interconnection

The common network access interface concept is favored by CCITT for network interconnection. In the CCITT model of packet networking, all networks offer the same interface to packet-mode subscribers and this is called X.25. X.25 is a virtual circuit interface protocol. However, gateways between networks employ an interface protocol called X.75 [33], which is much like X.25 but accommodates special network/network information exchange, such as routing information, accounting information, and so on.

Fig. 10(a) illustrates the basic network interconnection strategy proposed by CCITT. To appreciate the difference between this strategy and the "common subnetwork" strategy, it is necessary to have some understanding of the X.25 packet network interface. X.25 provides a virtual circuit interface for the setup, use, and termination of virtual circuits between subscribers of the networks. X.25 provides for flow control of packets per virtual circuit flowing into or out of the network. Subscribers may set up switched virtual circuits by sending "call request" packets into the network and receiving "call confirmation" packets in return. The standard also provides for permanent virtual circuits.

The public networks plan to employ X.25 interfaces; it can therefore be assumed that source and destination hosts in different networks will essentially want to exchange "call request" and "call accepted" packets through the mediation of one or more gateways. This strategy could result in a series of virtual circuits chaining source host to gateway, gateway to gateway, and gateway to destination host; alternately an end-to-end virtual circuit could be set up from source host to destination host, with the gateways acting as relays without any special knowledge of the virtual circuits passing across the network boundary.

The principle difference between the X.25 interface and X.75 interface is that virtual circuit setup and clearing packets are passed transparently by the X.75 gateway to the next gateway or destination. For reasons which are described below, it is necessary to maintain the sequence of packets belonging to a given X.25 virtual circuit as they pass through a gateway and enter the next network. Therefore, a virtual circuit is in fact created between the source host and intermediate gateway and between gateways. The X.75 gateway does not spontaneously generate any "call acceptance" packets in response to "call request" packets, but it does participate in the sequencing and flow control of packets on each virtual circuit passing through. Other differences between the X.25 and X.75 interface have to do with the nature of the internetwork accounting or routing information which might be exchanged over X.75 which would not be appropriate for a subscriber to exchange with the network over the X.25 interface.

The design of the X.75 type of gateway depends in principle upon all networks' use of the X.25 subscriber interface. Some networks, like the ETHERNET, cannot implement it without extensive modification, because there are no packet switches in the network to support the required packet reordering at the destination. The alternative is to insist that all internet applications rely on a sequenced data protocol built into the hosts or front-ends. For some services, such as packet speech, the potential overhead of resequencing packets before delivery to the destination may prevent the service from being viable. This problem could be amplified if packets are constrained to remain in sequence as they pass the X.75 boundary.

Fig. 10(b) and (c) shows variants of the CCITT interconnection strategy. In Fig. 10(b), we see an example in which only X.25 is used both as a network access method and as a means of passing traffic across network boundaries. A single subscriber or a pair of subscribers to two nets could interface to their networks via X.25 and to each other by means of some agreed and possibly private protocol.

Virtual circuits would be explicitly set up from source host to gateway, gateway to gateway, and gateway to destination host. The "internet" addresses of the source and destination hosts could be carried in the so-called "Call User Data Field" of an X.25 Call Request packet. This leaves the packet address field free to identify intermediate destinations (e.g., gateways), but preserves an ultimate internetwork source/destination address which the gateway can use to select the destination to which the next intermediate virtual circuit is to be set up.

An alternative to this is shown in Fig. 10(c) in which the subnets *A* and *B* use nonstandard virtual circuit interfaces, but agree to build gateway software employing X.75 signaling procedures across the gateway interface. This solution is substantially the same as that shown in Fig. 10(b), except there is now additional translation software in each gateway half to make each virtual circuit network-access protocol compatible with X.75 procedures.

There are some specific problems with the X.25/X.75 gateway strategy, which do not necessarily apply to other virtual call gateways [63]. The basic X.25 interface provides for the sequence numbering of subscriber packets mod. 8 or, optionally, mod. 128. Since X.25 is an interface specification, this numbering can only be relied upon to have local significance (i.e., host-to-packet switch). Some X.25 implementations use these host-assigned sequence numbers on an end-to-end basis. Others generate internal, network-supplied numbers to allow for repacking of subscriber packets into larger or smaller units for transport to the destination. If packet sequence numbers assigned by the source host were carried transparently to the destination without change, it might be possible to allow packets to flow out-of-order across the X.75 boundary to a gateway and thence into the next network. If the packet sequence numbers were still intact, they could be carried out-of-order to the next destination which might either be a gateway or an X.25 host. In the latter case, the original packet-sequence numbers could be used to resequence the packets before delivery. If the packets were being delivered to an intermediate gateway, they would not have to be sequenced there. However, the X.25 interface specification does not undertake to carry the host-supplied sequence numbers to the destination gateway or host in a transparent fashion, primarily so that the subnetwork can deal more freely with the physical packaging of the packet stream. For example, a source may supply packets of length 128 bytes while a destination may prefer to receive packets no longer than 64 bytes. To allow for such variations, the network must be free to renumber packets for delivery. These considerations have two consequences.

1) X.25 packet sequence numbers cannot be relied on for end-to-end signaling, though they could be so used if requisite information is known about the intermediate transit networks.

2) Packets must be delivered in sequence when passing to or from gateways and hosts on X.25 networks.

The second conclusion may be modified slightly. It is at least essential that packets be delivered in relative sequence on each virtual circuit. By maintaining independent sequence numbering on each virtual circuit, it is possible for hosts and gateways to refuse traffic on one virtual circuit while accepting traffic on another. There are two penalties for this. First, a gateway must keep track of which virtual circuits are passing through it. Second, dynamic alternate routing of packets belonging to the same virtual circuit through alternate gateways is not possible without resetting or clearing the virtual circuit. This last point is simply the consequence of not defining an end-to-end sequence numbering scheme, but instead relying on sequencing of the packets of a virtual circuit on entry to and exit from each intermediate network.

Some networks implement X.25 level acknowledgments (i.e., level 3) that have an end-to-end significance, but others make this purely a host-to-packet switch matter. As a conse-

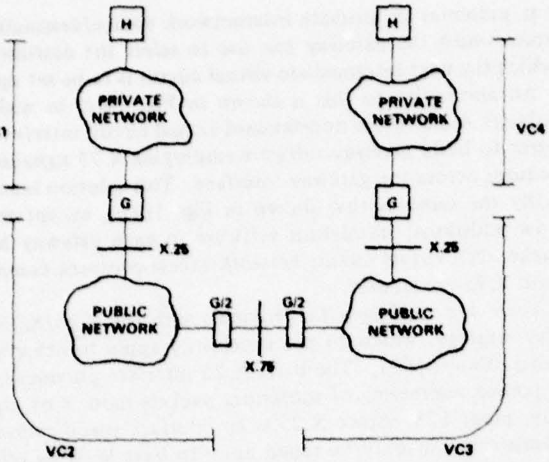


Fig. 14. Use of X.25 for public/private network interconnection.

quence, it is not possible to rely on X.25 packet acknowledgments to determine which, if any, packets were not delivered as a result of the resetting or clearing of a virtual circuit. Furthermore, even if a subnet were to offer an end-to-end acknowledgment between a source host and an X.75 gateway, this could not be assumed to guarantee that the acknowledged packet was delivered to the ultimate X.25 destination in another network.

X.75 is an interface intended for use between public networks. Thus, it is not likely to be used or even allowed as an interface between public networks and private networks. For the case illustrated in Fig. 14, X.25 interfaces could be provided between public and private networks (or other special interfaces) and X.75 interfaces between public networks. Consequently, gateways between public and private networks are likely to appear to be ordinary host computers in the view of the public networks.

The use of X.25 for private/public network interfaces and X.75 for public/public network interfaces leads to the situation shown in Fig. 14 in which an internetwork virtual circuit would have to be made up of several concatenated parts such as virtual circuits 1-2-3-4 (see also [52, Fig. 3.4]). Even if X.25 implementations uniformly permitted an end-to-end interpretation of packet sequence numbers and acknowledgments, there would still be separate virtual circuits required between the source or destination hosts and the gateways into the public networks. However, the concatenation of virtual circuits does not yield a virtual circuit. For instance, a gateway between the public and private net could acknowledge a packet but fail to get it delivered, in which case the subscriber will have been misinformed as to the delivery of the packet. This situation forces the end subscribers of private networks to implement end-to-end procedures on top of any concatenated virtual circuits provided by the public networks.

VIII. Practical Network Connections and Experiments in Progress

A number of networks have been connected successfully over the last few years. Most of these connections have been made in an *ad hoc* manner, using one of the following techniques.

1) One network is a star network with remote RJE and interactive stations. The other is a star or distributed network with clearly defined protocols. A device on the star network provides exactly the functions required by its own network on one side, and those of the other network on the other side.

2) Formal gateways are provided between the two networks, and protocol mapping occurs in the gateway.

3) A computer is a host on two networks. It is arranged that services are provided by accepting input from one network and putting it out on another, possibly after substantial processing.

4) Formal gateways are provided between the two networks. Sufficient agreement is obtained that end-to-end protocols (even high level ones) are common in the two networks. In this case, less activity is required in the gateway.

In the first method, a form of front-end computer is used. It has been adopted in the large airline and banking networks SITA [13] and SWIFT [14]. In each case the standards for the networks have been defined rigidly. SWIFT has even certified officially the devices of three manufacturers to provide interfaces to its network. The other side of the device is then programmed to meet the requirements of the star system being attached. In the two cases cited, only a simple message level of interface needed to be defined.

Other examples of the same technique are the connection of the Rutherford Laboratory (RL) star system [53] and the Livermore CTRNET to ARPANET. In these examples, more serious protocol mapping was required. ARPANET has a well-defined set of HOST-IMP, HOST-HOST, Virtual Terminal, and File Transfer protocols. All these had to be mapped into the appropriate procedures for the other network.

The second method has been applied only experimentally. The UCL interface between ARPANET and the UK Post Office Experimental Packet Switched Service (EPSS, [55]) and the National Physical Laboratory interface between EPSS and the European Informatics Network (EIN, [56]) are examples of this technique; a demonstration has even been made of EIN-EPSS-ARPANET with no extra problems encountered from the three networks being concatenated. Technically there is almost no difference between the first two methods. The second looks at first sight somewhat more general than the first, but almost the same problems have to be overcome. The difficulties come from the fundamental differences in the design choices made in the protocols of the different networks; these differences are in general difficult, and even sometimes impossible, to resolve completely. In the first method, they can sometimes be resolved using a specific facility in the star network; in the second, where two distributed networks are involved, this recourse may no longer be available.

One example of the problem occurs in the connection of EPSS and ARPANET. ARPANET can forward any number of characters at a time, and often uses full duplex remote echoing. EPSS works in a half-duplex mode, forwarding only complete records. A special "Transmit Now" has to be input by the user, and interpreted by the gateway, to ensure that partial records are forwarded. Another example, from the same application, occurs in File Transfer. ARPANET assumes an interactive process is live throughout the file transfer; all completion codes are passed over this live channel. The RL network (and EPSS) assume that file transfer is a batch process; they return network completion codes at a later time, and may delay acting on the commands. With the ARPANET-RL link [53], the file transfer job had to be given a very high priority, so that the completion code usually arrived before a timeout occurred; because of the nature of the way the computer was used for large real-time jobs, this did not always ensure that the job was run in a reasonable time.

There are several examples of the third technique. A DEC PDP 10 machine used on the Stanford University SUMEX project is a host both on ARPANET and on TYMNET; several machines at Bolt, Beranek and Newman are both on ARPANET and TELENET. Because the TENEX operating system has good facilities for linking between programs, it would be possible for interactive streams to come in one network and go out on another. File transfer problems would be simple in this configuration, because the hosts obey all the conventions, in any case, of each network. Of course, this mode of operation may require that files in transit between networks may have to be stored temporarily in their entirety in the host serving as the gateway between the networks.

The fourth technique is newer, and has many variations. As a result of agreement on the X.25, and partial agreement on the X.75, protocols, PTT networks are able to interconnect in a reasonably straightforward manner. The connections between DATAPAC and both TELENET and TYMNET have been done in this way. In each case, there has not been any agreement on higher level protocols, so the problems of host-host communication across concatenated networks is not resolved by these linkups of the subnets.

The ARPA-sponsored INTERNET project has tried to standardize to a higher level. A host-host protocol has been defined (TCP, [25]), and is being implemented on a number of different networks including Packet Radio [20], [21], ETHERNET [18], LCSNET [64] and the SATNET [22], in addition to ARPANET. This protocol is defined for use across networks; thus each packet includes an "Internet Header" which is kept invariant as the packet crosses the different networks. One aspect of the INTERNET program is to develop gateways which can interpret this header appropriately.

By late 1976, the ARPA project had connected together the Packet Radio Network, the ARPANET, and the Atlantic Packet Satellite Network using two gateways between the Packet Radio Network and the ARPANET and three gateways between the ARPANET and Packet Satellite Network. It is routinely possible to access ARPANET computing resources via either of the other nets and to artificially route traffic through multiple nets to test the impact on performance. In one such test, a user in a mobile van in the San Francisco area accessed a DEC PDP-10 TENEX system at the University of Southern California's Information Sciences Institute over the following path:

1) from van to the first gateway into ARPANET via the Packet Radio Network;
2) across the ARPANET to a second gateway in London, using a satellite link internal to the ARPANET;
3) across the Atlantic Satellite Network to a third gateway in Boston;
4) across the ARPANET again to USC-ISI.

The user and server were 400 geographical miles apart, but the communication path was 50000 miles long and passed through three gateways and four networks. Except for a slightly increased round-trip delay time, service was equivalent to a direct path through the ARPANET. Since the Packet Radio Network is potentially lossy, can duplicate packets, and can deliver packets out of order, the end/end TCP protocol was used to exercise flow and error control on an end-to-end basis. The availability of a common set of host-level protocols substantially aided the ease with which this test could be conducted.

The ARPA project also has high-level standard protocols already in existence to support file transfer and virtual terminals (the FTP and TELNET protocols [40]), and these are being retrofitted above the internet TCP protocol to provide a standard high-level internetwork protocol hierarchy.

IX. Regulatory Issues

The regulatory issues in the interconnection of packet networks takes a different form in North America than elsewhere. It is hard in a paper of this type to more than touch on some of the problems involved. The discussion here is simplistic in the extreme, and no attempt is made to put the issues in the legalistic language they really require.

In almost all countries the provision of long distance communication transmission and switching is provided by a regulated carrier. In most countries outside North America, this carrier is a single national entity—called the "PTT". In some countries (e.g., Italy) there are different carriers for different services—e.g., telegraph, telephone, intercity, international telephone, etc. In North America there are many carriers. Usually only one in each geographical area has a monopoly on public switched voice traffic. Also the so-called "Record Carriers" have some sort of monopoly on "record traffic," which is message traffic. In a "Value Added Network" (VAN), the operators rent transmission equipment from the carriers, and then add their own switching equipment. These VAN's are themselves regulated in what they may do, what traffic they may carry, and what rates they may charge. Between North America and Europe, specific "International Record Carriers" (IRC) have monopoly rights on data and message transmission—in collaboration with the appropriate European PTT's. The regulations take into account who owns the hosts and terminals, who owns the switches, who rents the transmission lines, what types of traffic is carried, what is the geographic extent of the network, and what is the technology of long distance transmission.

In Fig. 15, a single network N is sketched. It consists of switches S and transmission lines L; these together are called the data network, DN. It consists also of terminals T and hosts H; the exact difference between a terminal and a host is not very clear; we believe it is assumed that terminals mainly enter and retrieve data without processing; while a host transforms the information by processing. This definition probably does not meet the picture of modern "intelligent terminals," but it is always hard for the regulations to keep up with the technology. If the total network is all localized in one site, so that no communication lines cross public rights of way, then it can usually be considered from a regulatory viewpoint, as a single host in more complex network connections. The hosts and the terminals can be connected to the switches, and the switches to each other, either by leased lines, or by the Public Switched Telephone Network; the first type of connection is called a *leased* connection, the second *switched*. In the subsequent discussion of this section, the term "host" will include localized networks. In general we will assume the connections between the switches are via leased lines; if that is not the case, the regulations are much eased in general (though in some countries, like Brazil, no data transmission is permitted at all via switched telephone lines).

If all the hosts and switches are owned by one organization P, which also leases the lines, then P is said to own and operate the network, and it is called a "Private Network." There are

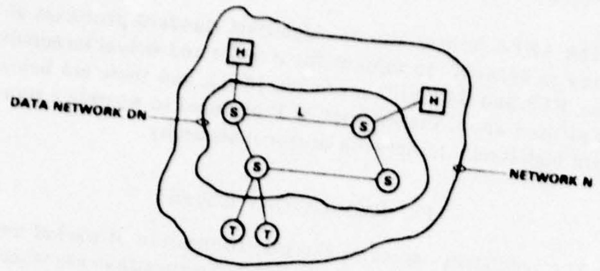


Fig. 15. Schematic of one network.

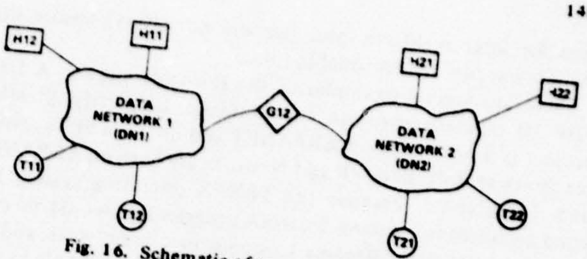


Fig. 16. Schematic of two connected networks.

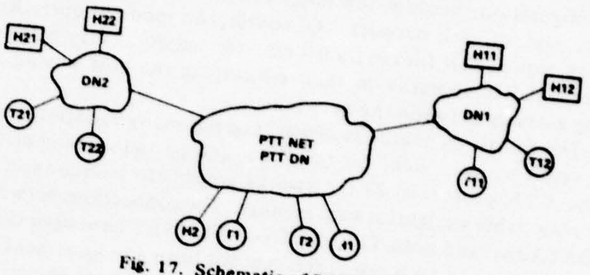


Fig. 17. Schematic of PTT model.

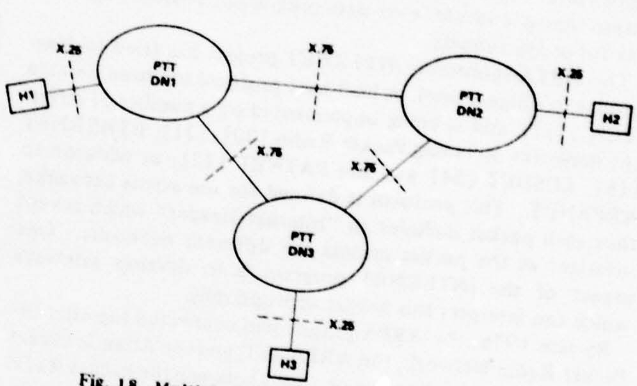


Fig. 18. Multiple PTT network interconnection.

minimal restrictions on such networks—though in West Germany, for example, higher tariffs are charged for the leased lines if any terminals or hosts are connected via the PSTN. In most countries such a network may not be used for the transfer of messages between terminals belonging to organizations other than *P*.

If the data network belongs to one organization, and the hosts to others, the data network is a VAN. Stringent regulations apply to VAN's, in most countries. With rare exceptions, in most European countries, VAN's can be operated only by the PTT's. In the U.S., they can be operated by other organizations, but only if approved as regulated Value Added Carriers (VAC's) by the Federal Communications Commission (FCC). One regulation imposed by the U.S. is that an organization operating as a VAC may not also operate a host for outside sale of services. For this reason, the companies TYMSHARE and ITT have had to spin off their VAC's into separate subsidiaries, TYMNET and ITT Data Services.

In the past, a few VAN's have been permitted to operate internationally for specific interest groups. Two such VAN's are SITA [14], for the airlines, and SWIFT [14] for the banking community. Here the regulations can be stringent. SWIFT has to pay specially high tariffs for its leased lines; its license to operate may be revoked when the PTT's can offer a comparable international service.

As soon as two networks, owned by different organizations, are interconnected, there are regulatory difficulties. This situation is illustrated schematically in Fig. 16. Even if one network is an internal one, so that it can be treated as a single host, its connection to other network immediately changes the latter's status. Thus in Fig. 16, the connection of *DN*1 to *DN*2 immediately changes *DN*2 to a VAN. In Europe it has been decreed that such private networks may not connect directly to each other, but only through a PTT network. Thus the most general configuration permitted by the European PTT's is illustrated in Fig. 17. Moreover, the PTT's have also agreed that only the X.25 interface will be provided to customers, though that interface was defined for the configuration of Fig. 15 rather than 17. The different PTT networks will themselves connect to each other by the different interface X.75 as illustrated in Fig. 18. This does *not* change, however, the interface seen by the private networks. Further work is needed to assess the suitability of X.25 in this role.

In the U.S., the regulations are not quite so stringent. Connections such as Fig. 15 are permitted even where one host belongs to a different organization than the network operator *P*—provided such connection is only limited and for the purposes of using the facilities of that network. This type of relaxation is really necessary, because of the difficulty of distinguishing between a "host" and a "terminal". In practice, in most countries, the line is drawn between leased line and PSTN connections. The former are usually not permitted without change of status of the network; the latter seem to downgrade the connection to that of a terminal.

The discussion above has treated the types of connections which can be made. In addition, the PTT's, and the FCC in the U.S., usually regulate the purposes for which the network can be used. In particular, there is a ban on such networks being used for message or voice transmission between organizations. How such measures are to be policed, gets us into another regulatory problem. For example the UK PO [57] has claimed a right to inspect the contents of any data message sent across lines leased from it; this right would be at variance with the privacy laws being enacted in many countries [58], [59]. This subject is a large one in its own right, and it is clearly beyond the scope of this paper.

Two other service problems will arise in international connections. First the impact and form of the privacy and transnational data flow regulations in different countries are different. Thus in the interconnection of international networks, a particular set of problems may arise, even when the appropriate regulations are obeyed in each network separately. Thus both Network 1 in country *A* and Network 2 in country *B* may obey their own national regulations. However when the networks *A* and *B* are connected, Network 1's practices may break country *B*'s regulations, and yet be accessible from country *B*. It is this class of problems which delayed seriously the permission by the Swedish Data Inspectorate Board for Swedish banks to connect their networks to SWIFT.

Secondly, some of the functions of networks or gateways legal in one country may be illegal in another. Thus U.S. carriers are not permitted to do data processing in their data networks; no such considerations apply in most European countries. Some of the protocol translation activities, some of the message processing activities, and some of the high-level services (e.g. the provision of multiaddress links) may well be classed as "Data Processing," and hence be illegal in the U.S. In interconnected networks, this raises the possibility that functions can be carried out outside the jurisdiction of the country in which the operator initiating the activity is sited, and yet which is illegal in that country. This subject is treated rather fully elsewhere [60]. A clear example of this is the use of message services operated by TYMSHARE and CCA on TELENET and TYMNET. While these services are legal in the U.S., their use by UK persons connected to TYMNET by the official International Packet Switched Service is clearly technically illegal; this use would contravene the UK Post Office Monopoly.

X. Unresolved Research Questions

There are many unresolved research questions; on some of them even the present authors do not agree with each other! Primarily these questions have a technical, policy, administrative, economic, regulatory, or operational aspect, or a combination of these.

One example of this is the question of the procedures to be used for internet routing. Here there are technical questions on what is feasible in view of the technologies used in the subnets; there are policy questions on when third country routing might be allowed; there are economic considerations on how much it would cost to do the necessary protocol translation to route through third countries, and on what charges the connecting transit network might make; there may be regulatory questions on which classes of data may flow through specific countries (related to the transnational data flow regulations); and there may be operational questions on whether in the event of failure in dynamic rerouting, reestablishment could take place with sufficient rapidity.

Among the outstanding research questions are, in alphabetic order, the following.

Access Control: What are the requirements and methods of implementation of access control? How should they affect internetwork routing?

Addressing: How should the International Numbering Plan, which goes to the level of known subscribers of public networks, be extended? Should this extension be in the numbering plan itself, or should additional user and network information be supplied? Should there be local, or only physical, addressing? Should there be internetwork source routing implied by the addressing?

Broadcast Facilities: What is the role of broadcast communication facilities in the provision of internet services? Should facilities using it be offered? Should technologies supporting it use it, particularly at gateways? What are the implications on protocols, especially with respect to duplicate and error detection?

Datagram versus Virtual Call Facilities: How should datagram and virtual call facilities be interconnected? How can one compare the relative performance and costs of the implementations? What criteria should be used in any comparison? When might datagram, or alternatively virtual calls, be desirable or essential between networks?

Data Protection: What are the effects of end-to-end data encryption on protocol translation?

Flow and Congestion Control: To what extent should one adopt congestion and flow control between gateways and their feeding networks, between gateways directly, or between gateways and the source? What are the relative effects of just discarding packets in gateways, and relying on the end-to-end protocol to detect and compensate for this? How is charging for discarded packets arranged?

High Level Protocols: There are still many questions on what should be standardized, and how rigid the standards should be. To what extent should the individual networks support common standards, and to what extent should protocol translation be feasible technically or attractive economically? What are the costs of maintaining standards or the economic advantages of standard hardware and software? How does the technology of individual networks and the proportion of internetwork traffic affect the decisions?

Internetwork Diagnosis: There are many technical problems in isolating faults in concatenated networks. There are also organizational and economic problems on who should be responsible for their repair, and how costs for service failures should be allocated.

Performance: How do choices of design parameters, and network services, affect the costs of the individual networks? How do the individual network performances and costs scale to large networks? How do the choices affect the feasibility, costs and performance of the gateways? How do the variations in technology or choice of parameters affect the performance in interconnected networks?

Routing Policies: To what extent and when should adaptive routing be used between networks? How can one recover from the partitioning of a single network, when there are still routes existing by going through other networks? How should administrative considerations affect routing policies between networks (privacy regulations, economic considerations of internet payments, desire to provide for high availability, etc.)? When is a hierarchical organization more efficient that a direct route search?

Services: What services are needed on an internetwork level? Clearly interactive and bulk transport services must be supported. What else is needed? Should the internetwork facilities be able to support voice, telemetry, and teleconferencing? What is the cost of supporting these latter services, and what is their effect on other facilities?

X.25 and X.75 and Related Recommendations: Is X.25 suitable for transaction processing? Are the present datagram proposals adequate? How should X.25 be extended for internet addressing? How should X.25/X.75 be modified to allow the connection of private to public networks, or private networks to each other? Do the X.3, X.28, X.29 pad concepts extend well to the internet environment, or should they be modified?

XI. Conclusions

In view of all the unresolved questions discussed in Section X, most of the conclusions which can be drawn in this paper must be tentative. From the early part of the paper, we have shown that it is essential that techniques be developed for connecting computer networks. Moreover, no single set of techniques will fit all applications.

The services which will normally have to be supported are terminal access, bulk transfer, remote job entry, and transaction processing. The quality and facilities of the services required will be very dependent on the applications.

The connections between networks can be made at the level of the packet switches or of hosts, and can be on a datagram or virtual call basis. Connection at the packet-switch level requires broadly similar network access procedures, or complex protocol transformation at the gateways between the networks. If the network protocols are different, interconnection can be most easily achieved if done at the host level. The higher levels of service can be mapped at service centers, which need not be colocated with the gateways–but very different philosophies of network services can be very difficult to map. Alternatively, subscribers can implement common higher level protocols if these can be agreed upon.

The principal problems in connecting networks are much the same as those in the design of the individual networks of heterogeneous systems–but the lack of a single controlling authority can make the multinet design problem more difficult to solve. It is essential to resolve the usual problems of flow control, congestion control, routing, addressing, fault recovery, flexibility, protocol standards, and economy. The public carriers have attempted to resolve many of these problems; particularly in the areas of flexibility, addressing, and economy we feel their solutions are not yet adequate. At the higher levels of protocol, much more standardization is required before we have really satisfactory long term solutions.

The advent of international computer networks, private networks which must communicate with other private networks (even if via public ones), and the new applications of computer networks, raise regulatory and legal issues which are far from resolution.

Many technical solutions to the problems of the connection of networks are discussed in this paper. Their applicability in view of the different technical, economic, and policy constraints imposed in different countries must still be assessed.

REFERENCES

[1] L. G. Roberts, "Telenet: Principles and practice," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, pp. 315-329, 1975.

[2] W. W. Clipsham, F. E. Glave, and M. L. Narraway, "Datapac network overview," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 131-136, 1976.

[3] J. Rinde, "TYMNET: An alternative to packet switching technology," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 268-273, 1976.

[4] R. E. Millstein, "The national software works: a distributed processing system," in *Proc. ACM Nat. Conf.*, Seattle, WA. 1977.

[5] A. Danet, R. Despres, A. Le Rest, G. Pichon, and S. Ritzenthaler, "The French public packet switching service, The TRANSPAC network," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 251-269, 1976.

[6] G. W. P. Davies, "EURONET project," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 229-239, 1976.

[7] R. Nakamura, F. Ishino, M. Sasaoka, and M. Nakamura, "Some design aspects of a public switched network," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 317-322, 1976.

[8] F. A. Helsel and A. J. Spadafora, "Siemens system EDS–A new stored program controlled switching system for telex and data networks," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 51-55, 1976.

[9] T. Larsson, "A public data network in the nordic countries," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 246-250, 1976.

[10] P. T. Kirstein, "Planned new public data networks," *Comput. Networks*, vol. 1, no. 2, pp. 79-94, 1976.

[11] P. T. F. Kelly, "An overview of recent developments in common user data communications networks," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 5-10, 1976.

[12] ——, "Public packet switched data networks," this issue, pp. 1539-1549.

[13] P. Hirsch, "SITA rating a packet-switched network," *Datamation*, vol. 20, pp. 60-63, 1974.

[14] G. Lapidus, "SWIFT network," *Data Communications*, vol. 5, no. 5, pp. 20-24, 1976.

[15] L. G. Roberts and B. D. Wessler, "The ARPA network," in *Computer-Communications Networks*, N. Abramson and F. Kuo, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1973, pp. 485-500.

[16] P. M. Karp, "Origin, development, and current status of the ARPANET," in *Proc. COMPCON 73*, San Francisco, CA, Feb.-Mar. 1973, pp. 49-52.

[17] L. Pouzin, "Presentation and major design aspects of the CYCLADES computer network," in *Proc. Third Data Communications Symp.*, Tampa, FL, Nov. 1973, pp. 80-85.

[18] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Commun. ACM*, vol. 19, no. 7, pp. 395-404, July 1976.

[19] A. S. Fraser, "SPYDER–A data communications experiment," *Comput. Sci. Tech. Report*, no. 23, Bell Laboratories, Dec. 1974.

[20] R. E. Kahn, "The organization of computer resources in a packet radio network," in *Proc. Nat. Computer Conf.*, AFIPS Press, pp. 177-186, May 1975.

[21] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," this issue, pp. 1468-1496.

[22] I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose satellite networks," this issue, pp. 1448-1467.

[23] D. Lloyd and P. T. Kirstein, "Alternate approaches to the connection of computer networks," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 499-504, 1975.

[24] L. Pouzin, "A proposal for interconnecting packet switching networks," IFIP Working Group 6.1, General Note no. 60, Mar. 1974.

[25] V. G. Cerf and R. E. Kahn, "A protocol for packet network interconnection," *IEEE Trans. Commun. Technol.*, vol. COM-22, pp. 637-641, 1974.

[26] C. Sunshine, "Interconnection of computer networks," *Comput. Networks*, vol. 1, 1977, pp. 175-195.

[27] CCITT, "Recommendation X.3: International user facilities in public data networks," *Public Data Networks, Orange Book*, vol. viii.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 21-23, 1977.

[28] CCITT, "Recommendation X.25: Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DEC) for terminals operating in the packet mode on public data networks," *Public Data Networks, Orange Book*, vol. VIII.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 70-108, 1977.

[29] CCITT, "*Provisional recommendations* X.3, X.25, X.28 and X.29 on packet-switched data transmission services," Int. Telecommunications Union, Geneva, Switzerland, 1977.

[30] CCITT, "Recommendation X.121–Int. numbering plan for public data networks," Study Group VII, Temporary Document 76-E, Int. Telecommunications Union, Geneva, Switzerland, April 25, 1978.

[31] L. Pouzin, "Virtual circuits vs. datagrams–Technical and political problems," in *Proc. Nat. Computer Conf.*, AFIPS Press, pp. 483-494, 1976.

[32] L. G. Roberts, "International connection of public packet networks," in *Proc. Third Int. Conf. Computer Communications*, Toronto, Canada, pp. 239-245, 1976.

[33] CCITT, "Recommendation X.75–Terminal and transit call control procedures and data transfer system on international circuits between packet-switched data networks," Study Group VII, Temporary Document 132-E, Int. Telecommunications Union, Geneva, Switzerland, Apr. 25, 1978.

[34] D. J. Farber and L. C. Larson, "The structure of a distributed computing system–The communication system," in *Proc. Symp. Computer Communications Networks and Traffic*, Polytechnic Institute of Brooklyn, pp. 21-27, Apr. 1972.

[35] P. Baran, "Broad-band interactive communication services to the home: Part II–Impasse," *IEEE Trans. Communications*, p. 178, Jan. 1975.

[36] R. E. Schantz and R. Thomas, "Operating systems for computer networks," *Computer*, Jan. 1978.

[37] L. Pouzin and H. Zimmermann, "A tutorial on protocols," this issue, pp. 1346-1370.

[38] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The interface message processor for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 36, AFIPS Press, pp. 551-567, 1970.

[39] S. Carr, S. D. Crocker and V. G. Cerf, "Host-Host communication protocol in the ARPA network," in *Proc. Spring Joint Computer Conf.*, vol. 36. Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 589-598, 1970.

[40] E. Feinler and J. B. Postel (Eds.), *ARPANET Protocol Handbook.* Network Information Center, SRI International, for the Defense Communication Agency, Jan. 1978.

[41] R. F. Sproull and R. D. Cohen, "High-level protocols," this issue, pp. 1371-1386.

[42] S. D. Crocker, J. F. Heafner, R. M. Metcalfe, and J. B. Postel, "Function-oriented protocols for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 40. Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 271-279, 1972.

[43] S. M. Ornstein, F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russel, and A. Michel, "The terminal IMP for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 40, Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 243-254, 1972.

[44] CCITT, "Recommendation X.21: General purpose interface between data terminal equipment (DTE) and data-circuit terminating equipment (DCE) for synchronous operation on public data networks," *Public Data Networks*, Orange Book, vol. VIII.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 38-56, 1977.

[45] CCITT, "Recommendation X.21-bis: Use on public data networks of data terminal equipments (DTE's) which are designed for interfacing to V-series modems," *Public Data Networks, Orange Book*, vol. viii.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 38-56, 1977.

[46] ISO, "High level data link control (HDLC)," *DIS 3309.2 and DIS 4335*, Int. Standards Org.

[47] V. Cerf, A. McKenzie, R. Scantlebury, and H. Zimmermann, "Proposal for an international end-to-end protocol," *Computer Communication Review*, ACM Special Interest Group on Data Communication, vol. 6, no. 1, Jan. 1976, pp. 63-89.

[48] A. S. Chandler, "Network independent high level protocols," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 583-602, 1975.

[49] —, "A network independent file transfer protocol," EPSS High Level Protocol Group, 1977.

[50] R. E. Kahn and W. R. Crowther, "Flow control in resource sharing computer networks," *IEEE Trans. Commun.*, vol. COM-20, pp. 539-546, 1972.

[51] L. Pouzin, "Flow control in data networks–Methods and tools," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 467-474, Aug. 1976.

[52] G. V. Bochmann and P. Goyer, "Datagrams as a public packet-switched data transmission service," Universite de Montreal, *Departement D'Informatique Report*, Mar. 1977.

[53] P. L. Higginson, "The problems of linking several networks with a gateway computer," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 453-465, 1975.

[54] J. Shoch, *private communication*.

[55] P. L. Higginson and Z. Z. Fischer, "Experience with the initial EPSS service," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 581-600, 1978.

[56] D. L. A. Barber, "A European informatics network: Achievements and prospects," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 44-50, 1976.

[57] B. Cross, "General license for message conveying computers," *London Gazette*, pp. 7662-7663, May 28, 1976.

[58] J. Freese, "The Swedish data act," in *Proc. Conf. Transnational Data Regulation*, Brussels, Belgium, ONLINE, pp. 197-208, 1978.

[59] R. Turn, "Implementation of privacy and security requirements in transnational data processing systems," in *Proc. Conf. Transnational Data Regulations*, Brussels, Belgium, ONLINE, pp. 113-132, 1978.

[60] A. R. D. Norman, "Project goldfish," in *Proc. Conf. Transnational Data Regulations*, Brussels, Belgium, ONLINE, pp. 67-94, 1978.

[61] E. Raubold and J. Haenle, "A method of deadlock-free resource allocation and flow control in packet networks," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 485-487, Aug. 1976.

[62] J. McQuillan, "The evolution of message processing techniques in the ARPA network," *Network Systems and Software*, Infotech State of the Art Report 24, Infotech Information Limited, Nicholson House, Maidenhead, Berkshire, England, 1975.

[63] P. Curran, "Design of a gateway to interconnect the DATAPAC and TRANSPAC packet switching networks," *Computer Communication Networks Group*, E-Report E-67, University of Waterloo, Canada, Sept. 1977 (ISSN 384-5702).

[64] D. D. Clark, K. T. Pogran, and D. P. Reed, "An introduction to local area networks," this issue, pp. 1497-1517.

4.3 Issues in International Public Data Networking

ISSUES IN INTERNATIONAL PUBLIC DATA NETWORKING

GARY R. GROSSMAN
Digital Technology Incorporated
Champaign, Illinois

ANDREW HINCHLEY
Department of Statistics and Computer Science
University College
London, England

CCITT Recommendation X.25 is the international standard interface to packet-mode public data networks (PDN's). CCITT Draft Recommendation X.75 defines the interface between packet-mode PDN'S. The interconnected PDN's will comprise an international public data network system. The interface standards and the system architecture that they imply will determine the kinds and qualities of service that will be available to subscribers. This paper discusses the interfaces and the implied system architecture in the light of subscribers' requirements and of the characteristics inherent in packet switching technology. A brief description of the X.25 and X.75 interfaces is followed by an evaluation of the "concatenated segment" architecture that they imply. An alternative architecture, based on the use of an end-to-end protocol within the international communication system, is presented and evaluated. This "DCE-DCE" approach seems to have technical advantages over the concatenated segment approach. The long-term economic implications need to be examined.

1. INTRODUCTION

CCITT Recommendation X.25(1) has been adopted as the international standard interface to packet-mode public data networks (PDN's). PDN's offering service through this interface are in operation or in construction in several countries, including the United States, Japan, Canada, United Kingdom, France, Germany, the Nordic countries, and Spain. The CCITT have drafted Recommendation X.75(2) to define the interface between packet-mode PDN'S. When the interface between PDN's has been adopted and implemented, the interconnected PDN's will comprise an international public data network system. The interface standards and the system architecture that they imply will determine the kinds and qualities of service that will be available to subscribers. This paper discusses the interfaces and the implied system architecture in the light of subscribers' requirements and of the characteristics inherent in packet switching technology.

2. SUBSCRIBERS' REQUIREMENTS

Subscribers require a unified international network system that meets their needs for both local and international data communication. The system must economically handle the kinds of traffic generated by most classes of subscribers. It must provide consistent interfacing procedures that are independent of the locations of the communicating subscribers. And it must provide acceptable throughput, delay, and error characteristics.

2.1 <u>Types of service</u>

Two types of service would fulfill the requirements of nearly all subscribers: virtual circuit service and transaction-oriented service.

2.1.1 Virtual circuit service. The majority of subscribers require a service which mimics the service currently provided by circuit-switched networks. In this service, communication is established and broken relatively infrequently, and communication is maintained for relatively long periods of time. Data must be delivered in order. The frequency of lost data and undetected bit errors must be within the range that the subscriber can tolerate. The rate at which data flows must be controllable. Service with these characteristics is called "virtual circuit" service.

2.1.2 Transaction-oriented service. A growing number of subscribers require a service which supports short transactions. In this service, the communication path need not be maintained between transactions. The transactions are independent of each other, and their order need not be maintained by the communications system. The subscribers' equipment is designed to handle errors; its correct operation does not depend on an error-free transmission medium. Data flow between any two points is relatively infrequent and therefore need not be controlled by the transmission medium. The communication system can discard data when the flow causes congestion. Service with these characteristics is called "transaction-oriented" service. Potential applications of this service are point-of-sale terminals, electronic funds transfer, distributed data base management, telemetry, and speech transmission.

Virtual circuit service and transaction-oriented service can be implemented in a number of ways. The way in which they are implemented can have a profound effect on many aspects of the communication system, as we shall see later.

2.2 Addressing

The way in which one subscriber addresses another through the public data network system should be the same whether the two subscribers are located on the same network or whether they are located on different networks. If a subscriber's "terminal" is a very complex one, such as a private network, the addressing scheme should permit high-resolution "subaddressing".

2.3 Routing

Packet switching technology provides very flexible means for routing calls and transactions between subscribers. In particular, the technology allows for alternate routing around failed or congested parts of the communication system, and even for dynamic rerouting without interruption of virtual circuits. The architecture of the communication system should take advantage of these features.

3. INTERFACES AND ARCHITECTURE

Fig. 1 shows the place of the X.25 and X.75 interfaces in the international public data network system. X.25 defines the interface between the subscriber's equipment (Data Terminal Equipment, or DTE) and the PDN's interfacing equipment (Data Circuit-terminating Equipment, or DCE). X.75 defines the interface between the internetwork interfacing equipment (Signalling TErminals, or STE's) of two PDN's.

3.1 Interface structure

Fig. 2 shows the three-level structure of both X.25 and X.75. For each interface, level 3 defines how packets control and effect the transmission of data between subscribers. For each interface, levels 1 and 2 define the physical characteristics and procedures, respectively, for reliably transmitting level 3 packets across the interface. CCITT are urgently studying the procedures for using multiple physical circuits between STE's to increase bandwidth and reliability.

3.2 Types of service

Both X.25 and X.75 currently provide for virtual-circuit service only. CCITT are currently studying the addition of transaction-oriented service.

The virtual circuits defined by X.25 and X.75 provide flow-controlled, order-maintained communication between the DTE's at each end of the connection. They also provide for the transmission of interrupts between the DTE's and for notification of error conditions, which may involve loss of data.

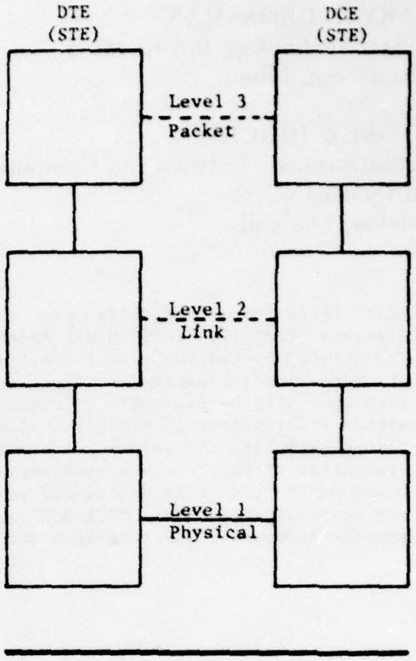


Fig. 2. X.25/X.75 Interface Structure

3.3 Virtual circuit structure

A virtual circuit between two DTE's that are connected to different PDN's consists of several concatenated components:

1. the interface, defined by Recommendation X.25, between one DTE and the DCE to which it is connected;

2. the mechanism, internal to the first PDN, that implements the virtual circuit between that DCE and the STE that connects the first PDN to the second PDN;

3. the interface, defined by Recommendation X.75, between the STE in the first PDN and the STE in the second PDN;

4. the mechanism, inrernal to the second PDN, which implements the virtual circuit between the second STE and DCE; and finally

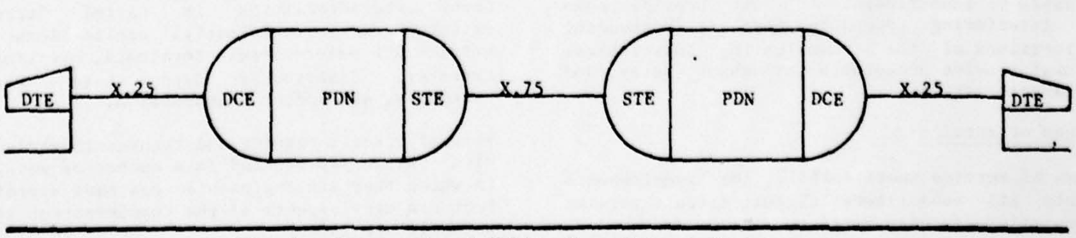


Fig. 1. International Public Data Network System

5. the interface, defined by Recommendation X.25, between the second DCE and the DTE that is connected to it.

If there are intervening PDN's, additional components consisting of an X.75 interface, an inter-STE mechanism, and another X.75 interface are added to the implementation of the virtual circuit for each intervening PDN.

It is important to note that the functions to be performed by the intranetwork mechanisms that connect DCE's and STE's are not defined by the CCITT.

3.4 Optional facilities

PDN implementors are not required to implement all of the facilities defined by the CCITT standards. Examples of these optional facilities are reverse charging and throughput class selection.

3.5 Parameters

PDN'S are permitted to vary certain parameters of the virtual circuits they implement on a network-wide, subscriber-by-subscriber, or circuit-by-circuit basis. These parameters include the maximum packet size permitted, whether reverse charging is supported, and so forth.

3.6 Addressing

The standards provide for a variable-length address that is divided into a network identifier and an intranetwork address. The maximum length of the intranetwork address (10 digits) is too short in some networks to provide for high-resolution subaddresses.

3.7 Routing

No procedure for internetwork routing is defined. There is no way for the STE in the originating PDN to specify the route that a call is to take. A means for determining the route of a completed call is defined, but there is no means for determining the route taken by an unsuccessful call or the point at which the call failed. If a virtual circuit is cleared due to a failure in some PDN along its route, there is no way for the originating STE to determine where the failure occurred. A unique call identifier is provided for each call. The call identifier is intended to be used in failure recovery, but no procedure for this is yet defined.

4. ADVANTAGES OF THE CCITT APPROACH

4.1 Flow Control

The CCITT approach provides flow control at every point along each virtual circuit's path. This permits the PDN's to effectively control buffer utilization and to prevent congestion.

4.2 Communication cost

The CCITT approach permits short packet headers at the expense of processing and maintaining status information. But costs of processors and memory are dropping faster than the costs of communication lines.(3)

4.3 DCE and STE implementation

X.25 and X.75 are very similar. Thus PDN implementors might be able to construct STE's by modifying DCE's.

4.4 Flexibility

The fact that the standards permit optional facilities and implementation parameters gives the implementors a great deal of freedom.

5. DISADVANTAGES OF THE CCITT APPROACH

The CCITT approach to public data network standards is limited by the concentration on interfaces, by the overall network structure that is implied, and by looseness of the standards.

5.1 Concatenated segment architecture

The architecture implied by the interfaces can be characterized as a concatenation of independent segments. The virtual circuit service, for instance, is implemented between two DTE's by, first, a segment between the the DTE and the DCE to which it is connected, followed by segments between the packet switches that make up the PDN, followed by a segment between the first PDN's STE and the second PDN's STE, etc. Each segment is a new interface. There is no aspect of the service which is defined as constant from one DTE to the other.

This approach requires that the entire virtual circuit service must be implemented in each segment. Every DCE and STE must handle flow control, error control, and all the other aspects of the service. Thus every DCE or STE must maintain state information about every virtual circuit that passes through it. And every DCE or STE must execute the relatively complex algorithms required to maintain the service.

The concatenated segment approach has the defect that any data loss along the path propagates through the chain with no means for transparent recovery. The undetected bit error rate for a virtual circuit is the sum of the error rates in each segment that comprises it.

5.2 Virtual circuit undefined

In the CCITT approach, the functional characteristics of the intranetwork segments of the architecture are left undefined. But, as fig. 3 shows, a virtual cir-

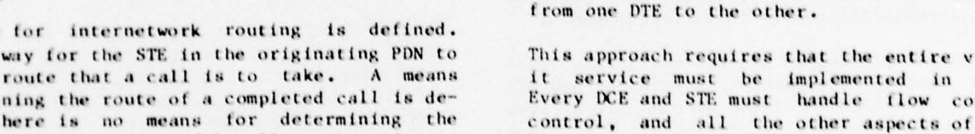


Fig. 3. CCITT Virtual Circuit Structure

cuit consists of concatenated segments, including the undefined intranetwork segments. Thus the functional characteristics of a virtual circuit cannot be defined from the definitions of its components and the relations between them.

5.3 Routing

The lack of a defined routing procedure, and of a means to determine where in the system a call or a virtual circuit has failed, make it impossible to reroute a call or virtual circuit around a failed element in the system. Again, the concatenated segment architecture makes it difficult to recover the state of the system that supports a virtual circuit, because the information that comprises the state of the system is distributed over all the elements on the circuit's path. If a recovery procedure using the unique call identifier is defined, this problem may be solved.

5.4 Reliability

The reliability of the system depends on the reliability of each of its elements. In other words, the system's reliability is equal to the reliability of its weakest element. It is for this reason that the implementors of the PDN's have gone to great lengths to construct extremely reliable processors and communication lines, usually via redundant configurations.

5.5 Loose standards

PDN implementors differ in their interpretation of the standards, and thus in the details of what they have implemented. For example, in X.25 the flow control mechanism is defined to have significance only at the DTE-DCE interface. The intranetwork flow control between DCE's is not defined. In fact, flow control is defined differently in different PDN's.

Some PDN's have implemented "synchronous" flow control: each unit of flow permission given by one DTE to its DCE is matched one-for-one with a unit of flow permission given by the other DCE to its DTE. Thus the PDN accepts data from one DTE only when the other DTE has indicated its readiness to receive. This places the burden of buffering to achieve throughput on the DTE'S.

Other PDN's have implemented "asynchronous" flow control: when a virtual circuit is established with a given throughput class, the PDN reserves enough internal buffers along the circuit's path to ensure that the given throughput is maintained. The PDN gives each DTE enough flow permission to fill the PDN's buffers. Thus the PDN relieves the DTE of some of the burden of buffering to achieve throughput.

When a DTE built for asynchronous flow control is attached to a PDN that provides synchronous flow control, the DTE may not have enough buffering to maintain the required throughput. When PDN's that employ different flow control schemes are interconnected, it is not clear what the result will be.

Thus, while implementation parameters and optional facilities permit flexibility, it remains to be seen how well differences in the values of these parameters can be made transparent to subscribers.

When subscribers connected to different PDN's that use different parameter values or different interpretations attempt to communicate, grave problems may result unless the the subscribers both employ a "least common denominator" interpretation of the standards. This approach may work for the subscribers. But it means that they must choose between on the one hand, foregoing the use of enhanced facilities that may be offered by their local PDN's, and on the other hand, employing different procedures on local and on internetwork virtual circuits.

Finally, it is difficult to see how a PDN that does not offer a given optional facility to its subscribers can serve as an intermediary between PDN's that do offer the facility.

6. AN ALTERNATIVE APPROACH

Earlier packet switching networks (ARPANET, CYCLADES, EIN) employed a somewhat different approach to providing services to their subscribers. This approach is called the "end-to-end" approach.

In the end-to-end approach, the network provides only a very basic packet switching service. Communication between subscribers and the network is on a packet-by-packet basis. If the subscribers desire a transaction-oriented service, they simply use the basic packet switching service as it is provided by the network. If the subscribers desire a virtual circuit service, they implement their own virtual circuits between them via an "end-to-end protocol". This end-to-end protocol uses the basic packet switching service as the underlying transmission medium. Virtual circuit establishment and termination, error control, and flow control are all handled by the subscribers at the ends of the communication path. Hence the name "end-to-end".

End-to-end protocols are designed with an imperfect data transmission medium in mind. Errors such as lost, duplicated, disordered, or altered data are handled by techniques such as retransmission and reordering. They thus require of the transmission medium only that it deliver a reasonable fraction of packets intact.

The end-to-end approach could be applied to the international public data network structure. As shown in fig. 4, the virtual circuits could be implemented via an end-to-end protocol between the DCE's at each end of the network communication path. This could be called a "DCE-DCE" approach. Intranetwork and internetwork switching would take place on a packet-by-packet basis. Some PDN's already work this way internally.

7. ADVANTAGES OF THE DCE-DCE APPROACH

7.1 Simple implementation

The DCE-DCE approach would make it unnecessary for every DCE or STE to implement the entire virtual circuit service for every virtual circuit which passes through it. Only the two DCE's at the ends of a virtual circuit would have to maintain state information about the virtual circuit. And only DCE's would have to execute the per-virtual circuit flow control and error control algorithms. STE's would simply pass packets between PDN's.

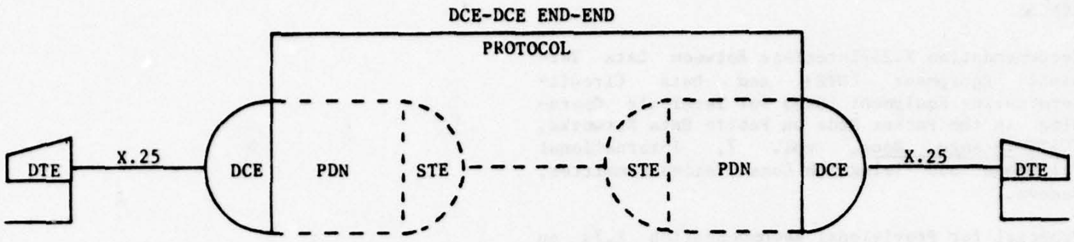


Fig. 4. Alternative System Architecture

7.2 Reliability

Failures along a path between the DCE's at the ends of a virtual circuit would only seriously affect the virtual circuit if all paths between the two DCE's became inoperative. Otherwise, packets which may have been lost through the failure need only be retransmitted by an alternate route.

7.3 Structure

The DCE-DCE approach would relieve subscribers of the burden of implementing all of the complexity of an end-to-end protocol, but would preserve the structural advantages of the end-to-end approach. The characteristics of the virtual circuit would be defined all along its path, as shown in fig. 5.

Another advantage of the DCE-DCE approach is that the international transaction-oriented service could be directly supported by the packet-switching mechanism internal to the system.

8. DISADVANTAGES OF THE DCE-DCE APPROACH

8.1 Complexity of protocol

End-to-end protocols would require somewhat more complex state information and algorithms.

8.2 Flow control

Flow control would not be provided at every point on a per-virtual-circuit basis. Other mechanisms would be required to prevent congestion and to control buffer utilization.

8.3 Communication cost

End-to-end protocols would require longer packet headers and thus higher communications costs.

8.4 Flexibility

The DCE's in every PDN would have to use the same end-to-end protocol. This would somewhat reduce the flexibility available to implementors of PDN's.

9. CONCLUSION

The adoption of an X.75 standard will have an even greater impact than that of X.25. It will set the design of the international packet-mode public data network system. The X.25 definition left open the question of concatenated segment versus end-to-end architecture. The current X.75 definition implies a concatenated segment architecture. This architecture requires extremely high reliability components that are correspondingly expensive. Thus the international public data network system may fail to achieve the economies that we have come to expect from packet switching.

The DCE-DCE approach is already in use within some PDN's. If it were extended to the international public data network system, the desired reliability could be achieved at a significantly lower cost for components. Which approach would give the best service to subscribers at the the lowest long-term cost needs to be determined.

ACKNOWLEDGEMENT

The authors are grateful to Vinton Cerf, Yogen Dalal, John Day, Peter Kirstein, and Carl Sunshine for their comments and suggestions.

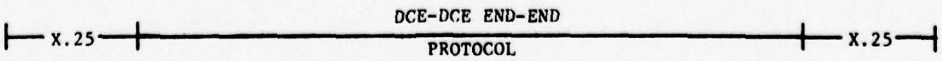


Fig. 5. Alternative Virtual Circuit Structure

REFERENCES

[1] Recommendation X.25/Interface Between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks, CCITT Orange Book, vol. 7, International Telephone and Telegraph Consultative Committee, Geneva.

[2] Proposal for Provisional Recommendation X.7x on International Interworking Between Packet Switched Data Networks, CCITT Study Group VII Contribution No. 109, International Telephone and Telegraph Consultative Committee, Geneva, December, 1977.

[3] L. G. Roberts, International Interconnection of Public Packet Networks, Proceedings of the Third International Conference on Computer Communication, Pramode K. Verma ed., Toronto, 3-6 August 1976, 239-245.

4.4 Supporting Transnet Bulk Data Transfer

Supporting Transnet Bulk Data Transfer

C. J. Bennett
Department of Statistics and Computer Science
University College London

ABSTRACT: This paper examines the problems of using existing transnetwork architectures to support transnetwork bulk data transfer. Two architectures are examined in some detail: the DARPA Catenet and the international public data network. Problems of integrating flow control techniques to support an end-to-end service are encountered with both architectures. An experiment is outlined to investigate these problems.

1. Introduction

Packet-switched data communication networks were originally developed to take advantage of the bursty nature of much of the interactive traffic exchanged between computers. It was quickly recognised that statistical multiplexing of data packets within a communication subnet was not in itself sufficient to handle the requirements of all users in a satisfactory fashion. Different applications have different requirements from the medium and some applications present different characteristics to it; for example, interactive transactions typically consist of packets generated at irregular intervals requiring responses within a fairly short time, while some telemetry data will be generated very regularly and may be able to stand a certain degree of data being lost in the process of the session. Hence, different techniques are required for handling the traffic in order to meet these constraints and to give acceptable performance to the application process.

This paper examines one particular class of applications, namely bulk data transfers requiring high amounts of throughput. The aim of the paper is to assess how effectively such transfers can be supported when they are being conducted across more than one network. Section 2 of the paper analyses the characteristics of bulk data traffic relevant to the discussion, and briefly surveys some of the techniques used within individual networks to handle such traffic efficiently.

Another paper submitted for this conference [13] examines some particular problems in transnet flow control in more detail using simulation. However, many factors influence the rates at which networks can accept and deliver traffic, and for this reason an architectural approach to flow control problems which identifies these factors and considers their interactions is also needed. This paper adopts such an approach. Section 3 is a general study of the areas of transnetwork connection which affect transnet flow, and considers both problem areas and the architectural framework within which solutions to these problems must operate. The next two sections are case studies expanding the arguments developed in Section 3. Section 4 assesses the DARPA Catenet, an approach to connecting networks which has been evolved by a research group associated with the Defense Advanced Research Projects Agency (DARPA). Section 5 looks at the internetwork architecture evolving in Public Data Networks (PDNs) as a result of the adoption of CCITT Recommendations X25 and X75.

In the light of the discussion up to this point, a transnet file transfer experiment is outlined in Section 6 which will incorporate both transnetwork architectures and will allow us to examine potential solutions to the problems encountered in a real transnetwork application. Finally, some conclusions are drawn on the current state of transnetwork flow control techniques.

2. Bulk Data Transfer and the Single Network

2.1 Traffic Characteristics of Bulk Data Transfers

Of the major network user services, bulk data transfer is the one with the most clearly defined set of requirements. The nature of the service is simply defined: it is to move a given body of data from point A to point B. The amount of data, its structure, even its content, is fixed (in most cases) at the time that the request for the transfer was made. Most applications requiring bulk data transfer are inherently very insensitive to timing constraints, unlike interactive conversations or many transaction applications, but do require that the structure and content of the data be preserved, i.e. that nothing is lost and that the original ordering is recovered at the destination. Because of this simplicity, most bulk data transfer implementations will adopt a very similar approach to packetising the data they are transferring so that efficient end-to-end flow is achieved. Therefore, the traffic can be characterised by the following two properties:

1) Nearly all the data packets involved in the transfer will be large; normally, as large as the network will accept.

2) Packets will be generated frequently.

The rate of packet generation is often subject to fairness criteria within the host operating system, and is also limited by the amount of traffic the network flow control procedures will allow. Nevertheless, one can state the following proposition: given that a data packet from a bulk data transfer arrives at some node in the network, there is a high probability that another such packet will follow very shortly after. This probability is affected by the factors mentioned, and is also higher for networks with fixed routing than for those with dynamic routing, but it will in almost all cases be significant.

Both properties are needed to define the characteristics a bulk data transfer presents to a network. In networks where the user pays by number of packets generated, line mode terminal conversations will be encouraged, in which case large packets may well be generated at relatively infrequent intervals. On the other hand, many telemetry applications may generate small packets quite often.

It has been noted that these properties are more suited to circuit-switched than packet-switched networks [15]. Where packet-switched networks are being used, it is important that they be able to optimise the throughput rates for bulk data transfer connections where short transfer times are desired. To achieve this, the network should fulfill a number of internal subgoals:

1) Network overhead in the packet should be minimised; the packets should be as large as possible, with only a small portion containing control information (packet headers and the like).

2) Packet loss rates through line errors and node congestion should be kept as small as possible, both on end-to-end and hop-by-hop bases.

3) A sufficient number of buffers should be allocated both at the ends and at intermediate points to avoid congestion and deadlocks, and to encourage smooth continuous traffic flow.

2.2 Support Techniques within Single Networks

The first of these conditions is very close to a restatement of the first characteristic of bulk data traffic given above; within the context of a single network it simply means that choice of packet size is the responsibility of the application program. We shall see below (Section 3) that the situation is not quite so simple when we come to consider multiple networks and transnet communication; here this condition can have a considerable impact on flow.

The second subgoal is a normal design requirement for communication networks in any case. The options were extensively discussed (at least for datagram networks) in [25]; although the possibility of offering different packet loss and error rates as a user-selectable option has been raised [19], these options have neither been studied fully nor implemented.

The third condition implies that the flow control techniques used in the network should be oriented towards the needs of the application traffic. Ideally, there should be some method of signalling to the network that a given traffic stream should be treated as bulk data. This is not always possible. Many networks, such as the ARPANET [1] place strict limits on the amount of buffer space that can be made available to a user connection within the network, and this is the same for all applications. To handle bulk data transfers within the ARPANET, the ALLOCation technique [24] was introduced; in this a destination node would automatically reserve some buffer space for future traffic from the source node for a short period of time if multi-packet messages were being transferred. However, this is not a general method for adapting automatically to the nature of data transfer traffic, as it relies on the fact that in the ARPANET packets for interactive conversations are [illegible] small. [illegible] [illegible] discussed above this is not always the case.

More recent datagram networks are building explicit awareness of user flow control requirements into the internal network architecture. The trans-Atlantic broadcast satellite network SATNET [20], for example, has a channel protocol which distinguishes between "block" and "stream" traffic, and will reserve slots in the transmission frames on either a burst basis or for a specified time period accordingly. Both the traffic type and various associated parameters can be made known to the satellite message nodes by the application process via a special protocol.

This trend is also observable in virtual circuit networks. X25 [4] provides a definition of user-selectable throughput classes, which represent an acceptance rate across an X25 interface. This may or may not be a guarantee, and does not necessarily carry any end-to-end implications, as the destination interface may support a completely different class for the connection, but it does commit the network to tying up buffers specifically for this connection. In the case of Datapac [29], the terminating network node (or DCE) will in fact commit more buffers to supporting the connection than the end host (DTE).

3. Transnetwork Flow Control Constraints

3.1 Differences Between Networks

When we require a service to be extended across several networks, we introduce new variables which will affect our ability to support the service efficiently. These all arise from the potential differences between networks, and many can affect the flow of traffic from source to destination. A comprehensive survey of the current problems in transnet connection is given in [8]; only those factors relevant to flow control will be studied here.

i) Differences in packet size: Networks can vary greatly in their internal packet sizes. X25 supports a range of maximum packet sizes between 16 and 256 octets, though 128 is recommended. Full ARPANET messages can be 1000 octets long, while some experimental, high data-rate networks such as the Cambridge University ring net [18] have packet sizes as low as 4 octets. These different sizes mean that larger packets have to be fragmented into smaller units before they enter the net with the smaller size; for efficiency they may have to be reassembled again before being delivered to the destination process if they can arrive out of order or contain an end-to-end checksum. The creation of these fragments, of course, means that each fragment will carry its own addressing information and will generate its own network overheads - acknowledgements, retransmissions etc. In particular, buffers will be required for the creation of the fragments and for reassembling them into their original units. Thus the way in which fragmentation and reassembly is implemented can directly affect the rates at which the end processes produce and consume buffers.

ii) Differences in service characteristics: The differences between adjacent networks caused by factors such as the networks' communication media, geographical size, or mobility, and visible in such characteristics as throughput rates, end-to-end delay, and error and loss rates can all affect transnet flow and congestion control schemes.

iii) Differences in user options: The networks may be virtual circuit or datagram networks, and may offer different degrees of reliability. Datagram networks may not preserve ordering, in which case the gateway or the end process may have to reserve more buffers than would otherwise be necessary. Adjacent networks may or may not offer user-controllable options to select various grades of service, which may differ between adjacent networks. More subtly, they may conform to a common specification but differ significantly in the details of the implementation, as in some X25 networks.

3.2 The Role of Gateways

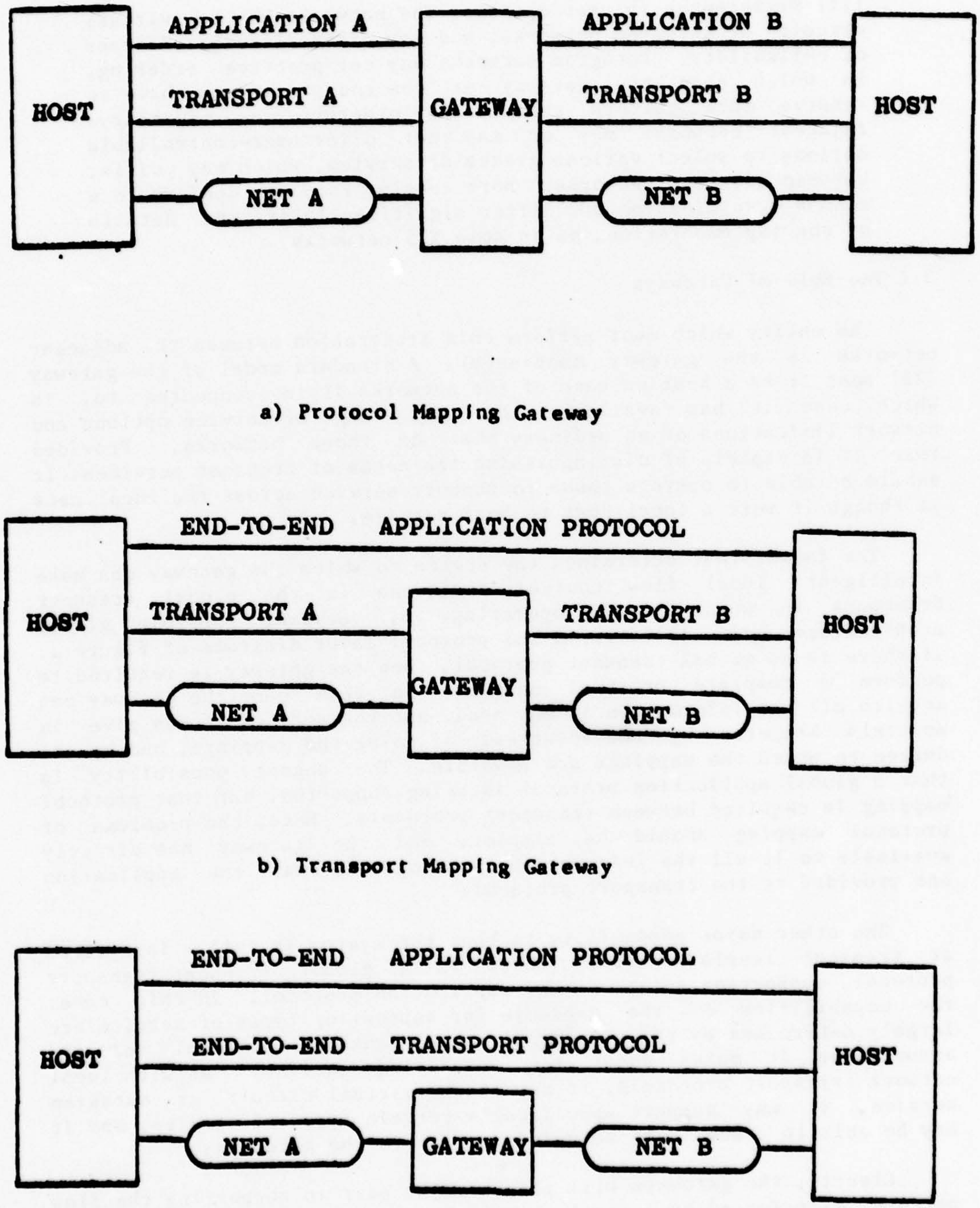
The entity which must perform this arbitration between the adjacent networks is the gateway machine(s). A standard model of the gateway [22] sees it as a host on each of the networks it is connected to, in which case it has available the full range of service options and network limitations of an ordinary host on those networks. Provided that it is capable of distinguishing the needs of transnet services, it should be able to operate these to support service across the local nets as though it were a local host to host service.

The factor that determines the degree to which the gateway can make intelligent local flow control decisions is the global transnet framework in which it is operating in. Several possible global architectures are illustrated the protocol layer diagrams of Figure 1. If there is no global transnet protocol, then the gateway is required to perform a complete protocol mapping; in this case the gateway can acquire all the information it may need, but the service it can give is severely hampered by the overhead of doing the mappings, and by the degree to which the mappings are possible. The second possibility is that a global application protocol is being supported, but that protocol mapping is required between transport protocols. Here, the problems of protocol mapping should be simpler, and the gateway has directly available to it all the information about options that the application has provided to the transport protocol.

The other major possibility is that the system is fully integrated at transnet levels - i.e. there is a global transnet transport protocol, supporting an end-to-end application protocol. In this case, the capabilities of the gateways for supporting types of service are largely determined by the nature of the transport protocol and the assumptions it makes about the underlying service. As with local network transport protocols, it may assume virtual circuit or datagram service, it may support specifically certain types of service, and it may be able to communicate such information to the gateways.

Clearly, the gateways will play a major part in supporting the flow pattern experienced by a particular application. It also becomes clear that the subgoals we defined in Section 2 are no longer entirely adequate for defining transnet support procedures at gateway level. Bearing in mind the differences between networks discussed above, we can redefine these subgoals for gateways as follows:

i) Support maximum size fragments with minimum overhead: the gateways should minimise fragmentation, and in certain cases may perform intermediate reassembly.



a) Protocol Mapping Gateway

b) Transport Mapping Gateway

c) Network Access Gateway

Figure 1: Fundamental Transnet Architectures

2) Minimise packet loss: The gateways should minimise packet loss at the gateways and choose reliable delivery options in local networks. The local network should be able to handle loss due to packet errors; losses due to congestion may occur in gateways unless proper congestion control techniques are instituted.

3) Optimise buffer availability: The gateways should relate the number of buffers assigned to this application to the expected demand. It should also be able to select user options from the local network on the basis of the applications requirements. The end-to-end transport mechanism, where applicable, should provide the gateways with some means of obtaining this information.

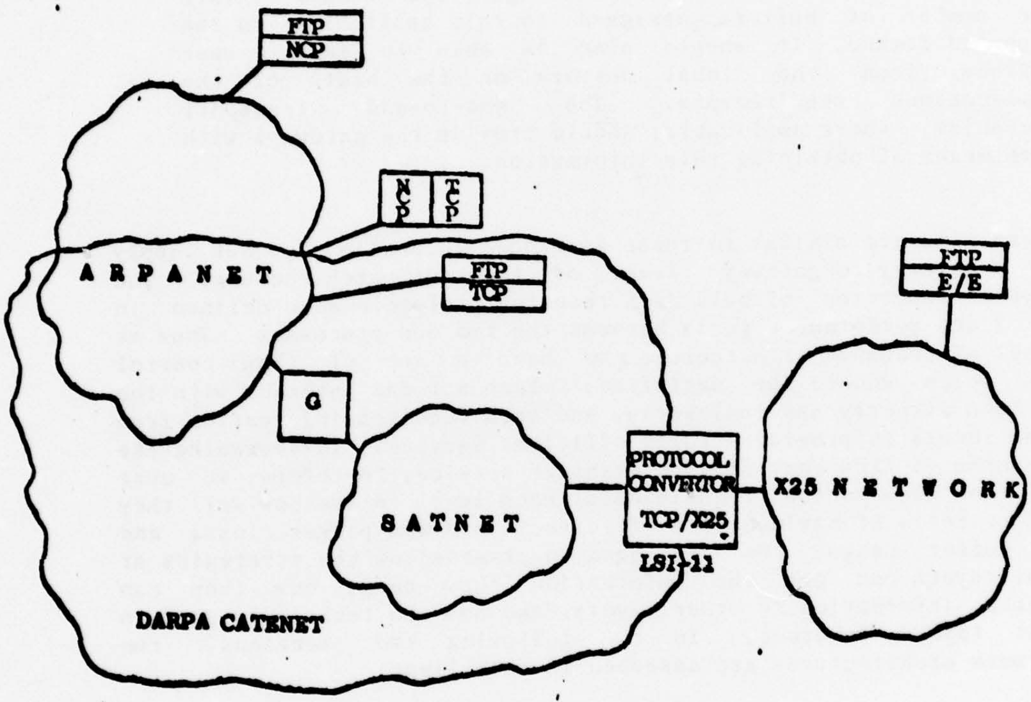
These aims are similar to those developed in Section 2, but apply to the gateway-to-gateway level of transnet architecture. The 'archetypal properties' of bulk data transfer protocols also defined in Section 2 are performance goals between the two end processes. Thus at each layer of transnet architecture we have a set of flow control criteria which should be satisfied. Each set can interact with the others, both directly and indirectly, and each requires information from the other levels to provide a fully efficient service. In assessing the effectiveness of flow control in a transnet service, therefore, we must examine the flow control techniques at each level to see how well they meet the criteria of maximum data efficiency, minimum packet loss, and optimal buffer usage. We also need to examine how the strategies at different layers can get the information they need, how they can communicate information to other layers, and how the techniques used in different layers interact. In the following two sections, two transnetwork architectures are assessed in this light.

4. File Transfer in the DARPA Catenet

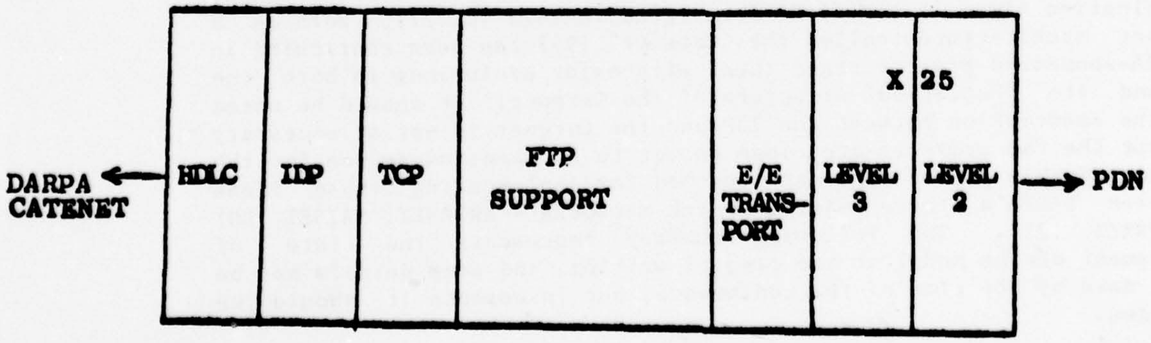
4.1 The DARPA Catenet Architecture

The Transmission Control Protocol, or TCP, is a network-independent transport protocol which has been proposed as a vehicle for transnet communication since it was first formulated in 1974 [6, 27]. Work on a transnet architecture (called the 'Catenet' [9]) has been continuing in a DARPA-sponsored project since then, with major evolutions in both the TCP and the theoretical structure of the Catenet. It should be noted that the association between the TCP and the Catenet is not a necessary one, but the two projects are close enough to be regarded as one for the purposes of this paper. The main testbed for implementing these ideas has been DARPA's three major research networks - ARPANET, SATNET [20] and PRNET [21]. The following summary represents the state of development of the model at the time of writing, and some details may be out of date by the time of the conference, but in essence it should be unchanged.

Figure 2 shows the basic model behind the DARPA Catenet [33]. Currently the Catenet consists principally of the three DARPA-related networks given above, and gateways have successfully been built between ARPANET and SATNET [20] and ARPANET and the PRNET [21]. Other networks will be added in the future. The networks within the Catenet are



a) The Future UCL Configuration (simplified)



b) Protocol Structure of a File Transfer 'Meta-Gateway'

Figure 4: An Experimental Configuration for Transnet File Transfer

connected by gateways which transfer packets from one net to the next on a datagram basis. An internet packet may take any route through the Catenet, it may be fragmented by the gateways at any time, and each of the fragments themselves may take any route, to be reassembled at the destination host before delivery to the destination transport process. The addressing and fragmentation procedures supporting this minimal set of facilities is defined by the Internet Datagram Protocol [28] - or IDP. The routing procedure is currently based on fixed routing tables, but it is shortly to be replaced by one using an adaptive algorithm [31]. The IDP makes no guarantee of reliable or sequenced delivery, and hence the gateway congestion control currently consists of a simple blocking technique [7]. It does, however, provide a number of service options, which are still undergoing refinement. The oldest one is a "Don't Fragment" option; others adopted recently include the selection of qualitative delay classes (of the type "As fast as possible", "Fast", "Normal", "Don't Care"), and the creation of reliability options to facilitate transmission over very lossy networks.

Above the IDP at the source and destination hosts comes various internet transport protocols. The most completely defined, and the most well known of these is TCP, which will be at the basis of most conventional network services such as virtual terminals and file transfer. Other more specialised transport protocols such as a Real Time Protocol (RTP) are envisaged for particular applications such as packet voice, but these will not be considered further in this paper.

4.2 End-to-End Flow Control

TCP provides reliable sequenced delivery of almost arbitrarily large internet packets. The end-to-end flow control used by TCP is based on a windowing scheme using octet-based sequence numbers but incorporating information on the buffer sizes exchanged at set-up time to ensure smooth changes in window size [3].

Clearly, when small windows are in use, the scheme approximates to the sequenced packet-by-packet acknowledgement schemes required for handling interactive traffic, and which have been in existence for many years. When large windows are used, the situation changes, and acknowledgements for large amounts of data need only be generated at infrequent intervals - which makes windowing schemes insensitive to out-of-order deliveries. Windowing, then, comes into its own as a technique in precisely the kind of situation we encounter in bulk data transfer, and so the studies which have been made of various TCP flow control policies have a particular significance for bulk data transfer.

The window size sent by the receiver to the sender represents a promise of buffer space availability at the receiver's end. Two policies have been identified on which this promise can be based [32,14]. The 'conservative' policy uses the window size as a firm guarantee of buffers actually available to the receiving TCP at the time it generated the message. The 'optimistic' policy bases window size calculations on some predictive heuristic to give a reasonable estimate of what the receiving TCP expects to possess in the near future. As one would expect, simulation studies [11] have shown that the conservative policy is the best one to follow, especially when buffer space management is the responsibility of the user process, which is the policy encouraged by the TCP implementors. In this case, an optimistic policy must try to gauge the demands that the user process is going to make on the TCP and the resources that it will make available. The current suggestions for TCP user interfaces do not provide a means for doing this under user control, although such an interface could be designed without violating the protocol. End-to-end congestion control is based on retransmission and positive acknowledgements. Retransmission policies have been investigated in [12]; this study favours a policy of increasing the retransmission timeout between retransmissions, to avoid loading the Catenet with retransmissions.

4.3 Gateway-to-gateway Flow Control

Currently, gateway-to-gateway flow control is non-existent: if a gateway cannot accept a packet, it is simply dropped. Since the packet is an internet datagram, its eventual arrival at the destination will depend on the end-to-end retransmission of the packet. This policy is acceptable for interactive traffic, as one can assume that the gateways will have sufficient time to clear the source of congestion, or cause a routing change, before the next packet arrives. However, it is not an acceptable proposition for bulk data traffic, since, from the property we established in Section 2, the discarding of one packet due to congestion will probably lead to the discarding of several subsequent packets before the situation is corrected, possibly involving a large amount of data. Congestion is more likely to be caused by bulk data transfers than by other applications, but the advance buffer reservations that could reduce it by anticipation cannot be made as there is no way of indicating what may be required.

In practice, the situation may be even more complicated than this. While gateway-to-gateway delivery may not be reliable, the local network may guarantee reliable delivery. Because there is no gateway-to-gateway acknowledgement, a gateway may be unable to distinguish congestion within the network from congestion in the neighbouring gateway, and it certainly cannot tell which packets are being delayed because of it. Thus the situation can arise whereby a gateway is injecting an end-to-end retransmission into a network which is still trying to deliver the earlier transmissions. This situation has in fact been detected in TCP experiments across the ARPANET [2].

It is not clear that the introduction of a reliability option is a solution to this problem, as the essential features of the solution are that the gateway must be able to detect that it has handled an earlier transmission of the packet, and that it knows why the earlier transmission did not reach the neighbouring gateway. Both features require fixed routing, and the first requires that there is a defined coupling between the end-to-end sequence numbers and the fragment numbers defined in the IDP for use in the gateways; but routing is dynamic, and no such coupling has been defined.

Very recently, it has been proposed that a gateway detecting congestion should generate an advisory message requesting the quenching of traffic at the source. Such a policy could alleviate many of the problems described here, but it raises additional problems of fairness, and of the action to be taken if such messages are ignored.

4.4 Effects of Fragmentation

Another aspect of gateway-to-gateway functions which concerns us here is fragmentation. Fragmentation is a procedure which is more likely to be applied to bulk data transfer packets than to those from other applications because of the large packet sizes involved. However, the overheads of fragmentation are large. Each TCP packet has at least 16 octets of header and each internet fragment will have at least a further 16, in addition to any local network headers that must be generated for each fragment. A maximum efficiency of 77% is possible on the ARPANET for internet fragments contained within a single ARPANET packet [28], and this figure drops to 63% for the first fragment of the packet, carrying the TCP header.

The disparity in maximum packet sizes affects the processing efficiency of destination hosts in more subtle ways. A reasonable buffering strategy at a receiving host in a bulk data transfer would be to allocate buffers of a size fixed by the maximum packet size of the local network. Similarly, a reasonable fragmentation policy is to always create fragments of a maximum size where possible. However, the actual pattern of fragment sizes created by this policy may render the buffer consumption very inefficient. For example, a 256 octet internet packet going from a 256 octet network into a 128 octet network under this scheme will be fragmented into three packets - the last containing 16 octets of header and 16 octets of data. If this were happening in a bulk data transfer, then the receiver would be processing incoming packets three times as fast as the source is producing them, and a third of the packet buffers would be largely empty. This renders the receiver very sensitive to high processing loads in its host, directly affecting end-to-end throughput.

Such effects can occur even without fragmentation. In the example above, a file being transferred in the reverse direction will have 256 octet buffers allocated to it by the receiver, and every one of these will only be half filled.

Again, because the routing is dynamic and the communication datagram based, it is difficult to see any algorithm guaranteeing to avoid these effects in the DARPA Catenet. Intermediate reassembly is not an option supported within the IDP, although it is recognised that in some cases a private gateway-to-gateway scheme may be necessary [30]. This is a direct result of supporting dynamic routing. If intermediate reassembly was implemented, each gateway would then have to delay each incoming fragment long enough to determine whether neighbouring fragments were taking the same route and whether reassembly was possible. Reassembly is not guaranteed, but the potential for increasing congestion and adding to end-to-end delay is high.

A second possibility is to use the "Don't fragment" option, relying on the fact that intermediate networks are required to support segments of at least 576 octets. This is a disguised form of the previous approach, as in order to support it many networks would have to implement a private fragmentation/reassembly protocol in the gateways adjacent to that network. This may in fact cause reassembly to take place even in situations where it can give the user no benefit, as the packet has to be fragmented again in the next network. It does have the great advantage, however, that the two end processes can efficiently adopt a common large buffer size for bulk data transfers.

4.5 Gateway to Local Network Flow Control

Finally, we consider the use that gateways can make of services available in local networks. For bulk data transfer, we wish the gateways to make requests indicating high throughput traffic. As we saw in Section 2, local nets typically do this by allocating some proportion of their resources to the requirements of the transfer. However, we again come up against problems due to the datagram nature of the Catenet - although the gateways may request bulk transfer facilities from a local net, they may not in fact be able to deliver the bulk to transfer. The ALLOCation technique used in ARPANET to support file transfer (see Section 2) relies on a pipe-lining effect for its efficiency; it delivers a new reservation with each acknowledgement. If a packet is late reaching a gateway, or it enters the ARPANET via a different gateway then this effect will be lost, the reservation at the destination node will be cancelled, and the source node will have to reestablish the reservation before the next message can be delivered.

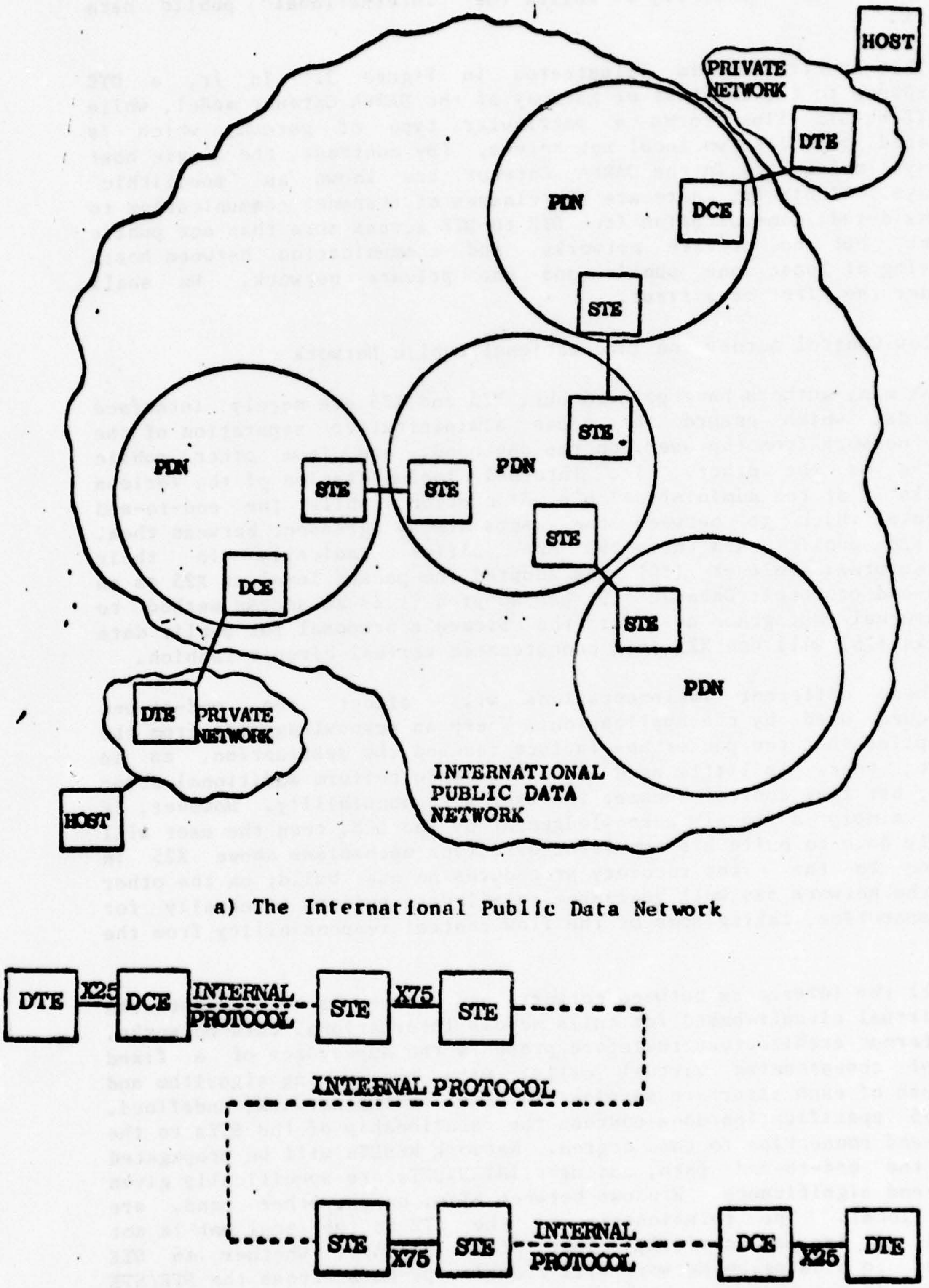
In other networks, the effect of late arrival or temporary rerouting is more likely to be seen as a burst of traffic, possibly causing congestion problems inside the networks. This sort of effect will be seen in satellite networks using reservation schemes for channel access. Since gateways are unable to exploit the end-to-end characteristics of the traffic to indicate the parameters which should be used to control the channel algorithm for it, they can only select these parameters on the basis of the packets immediately available in the gateway queues. A gap in the flow of traffic will cause loss of the channel, which will take at least a quarter of a second to recover regardless of the reservation scheme used. In this time many packets could arrive at the ground station, and although large reservations can be made, they may not be granted in full if the channel is heavily loaded. Thus it can take some time before the buildup is reduced to a 'normal' level.

In short, the DARPA Catenet approach creates a large number of flow problems for supporting transnet bulk data transfers which are still being resolved. This is primarily due to the minimal, datagram nature of the Catenet model. The end-to-end flow control techniques used by TCP, on the other hand, are well suited to file transfer, and may well compensate for the internal inadequacies.

5. Bulk Data Transfer Across Public Data Networks

5.1 The International Public Data Network

CCITT Recommendation X25 [4] has laid down an interfacing standard for the connection of private data terminal equipment (DTEs) to the public data networks that will become available shortly in many industrial countries. Briefly, X25 consists of three levels, defining the physical, link level and packet level interfaces between the DTE and the network's data communication equipment (DCE). The packet level, which concerns us here, defines procedures for the establishment and breaking of virtual calls across the interface, using a windowing technique for flow control in which the window size is fixed and represents numbers of packets transferred rather than volume of data. The impending adoption of draft Recommendation X75 [5], which specifies a similar interface to be adopted by switching terminal equipment (STEs) to connect adjacent public data networks, brings into being the basis of an internet architecture which is of considerable importance to many data communication users. Since in most countries (outside North



a) The International Public Data Network

b) The End-to-end Virtual Circuit

Figure 3: Data Transfer Across the International Public Network

America) there will be at most one public data network, the internet architecture can reasonably be called the "International" public data network.

This architecture is illustrated in Figure 3. In it, a DTE corresponds to the end host or gateway of the DARPA Catenet model, while the STE-to-STE link forms a particular type of gateway which is separated into its two local net halves. (By contrast, the single host gateways implemented in the DARPA Catenet are known as 'monolithic' gateways.) Within it, there are two classes of transnet communication to be considered: communication from DTE to DTE across more than one public network, but no private networks, and communication between hosts involving at least one public and one private network. We shall consider the first case first.

5.2 Flow Control Across the International Public Network

As many authors have pointed out, X25 and X75 are merely interface standards, which ensure the clean administrative separation of the public network from the user, on the one hand, and from other public networks on the other. The internal implementation of the various networks is at the administration's discretion, while the end-to-end protocols which go between the users are by agreement between them. Thus X25 public data networks can differ radically in their architectures: Telenet [10] has adopted the packet level of X25 as an end-to-end protocol; Datapac [29] has adopted it as an access method to an internal datagram network; the Siemen's proposal for public data networks [26] will use X25 in a concatenated virtual circuit fashion.

These different implementations will affect the end-to-end techniques used by the application. Where an acknowledgement from the DCE implies that the packet has in fact reached the destination, as in Telenet, there is little need for the user to perform additional error checks, but flow control becomes the user's responsibility. However, if it is simply a local acknowledgement by the DCE, then the user will probably have to build his own error detection mechanisms above X25 in addition to the error recovery procedures he must build; on the other hand, the network may well guarantee to allocate buffers internally for the connection, taking some of the flow control responsibility from the user.

All the interfaces between networks and between users and networks are virtual circuit-based for calls across international data networks. The internet architecture therefore presents the appearance of a fixed path of concatenated virtual calls, with the routing algorithm and structure of each alternate section, internal to the network, undefined. The X75 specification does address the relationship of the STEs to the end-to-end connection to some degree. Network RESETs will be propagated along the end-to-end path, and user INTERRUPTs are specifically given end-to-end significance. Windows between STEs, on the other hand, are purely local. The relationship of the STE to the local net is not defined. It is not strictly clear, for instance, whether an STE attached to a datagram network will transfer packets across the STE/STE interface in the order in which the user created them or in the order in which they arrive. If the latter choice were made, the destination process would have to reserve buffers to handle misordered packets, or to force end-to-end retransmissions should misordered packets arrive.

Problems of this nature are fairly easy to overcome by making sensible design choices; in the example above, that the STE will transfer packets in the user ordering. More subtle problems arise with end-to-end error control and flow control. Any packet loss anywhere in the path occurring, for instance, because of a RESET between two STEs, will be propagated down the rest of the path. The packet loss will merely be compounded by the requirement of X75 that the RESET should also be propagated down the rest of the path. Losses occurring for other reasons may be propagated to the end of the path without detection.

Most importantly, unless X25 acknowledgements have purely local significance in the DTE-DCE interface, their significance across more than one public net is completely indeterminate: they may represent acceptance by the first STE, by the remote DCE or DTE, or by any intermediate STE. Hence the user cannot regard international calls as having anything other than local flow control. Because of the potential differences in flow control strategies within public nets, the user cannot make any assumptions about the buffer reservations that will be made for his connection outside his local public network.

Clearly, this situation has to be remedied by the introduction of an end-to-end protocol across the public data system. It has been suggested [16] that the end-to-end protocol should be implemented between source and destination DCEs. It is highly unlikely that this solution will be adopted, as it requires all public data networks to adopt the same end-to-end protocol, which will compromise the independence of the administrations to modify their internal architecture at will. It is far more likely that appropriate DTE-to-DTE protocols will have to be implemented by the groups of users who wish to communicate.

5.3 Flow Control Across Constituent Public Networks

Given that a global architectural framework has to be imposed above the public data networks, to provide end-to-end reliability, ordering, and some flow control technique, we can now consider how well individual public data nets can support transnetwork bulk data transfers. X25 and X75 provide several user facilities which are of great assistance here, but they also have several defects which arise from the lack of specification of the end-to-end significance of the architecture. Window sizes are only defined across the interface, and there is no guarantee that a request to reserve large buffers will be carried from DCE to STE or from STE to STE; however, STEs are required to transmit user throughput class requests unchanged. The significance of these can vary from net to net: one net can assume that they represent a guaranteed rate, the next that they represent a maximum rate and the next that they represent a minimum rate. In any case, they only represent throughput across the interface, and cannot be used as a basis for guaranteeing high throughput across a local public network.

Nevertheless, user control of flow rates between STEs across local networks is greater than it is between gateways in the DARPA Catenet. A means of requesting high bandwidth services does exist, and because of the concatenated virtual circuit nature of the path, one can guarantee that all the packets involved in the transfer will take that path. Public data nets relying on short-term pipelining effects of the sort mentioned in Section 4 will have a greater chance of keeping the pipeline full than the DARPA Catenet does.

X25 and X75 provide a "More Data" option to indicate a sequence of related packets. X25 also supports a number of possible packet sizes, all powers of 2 (with the exceptional case of 255 octet packets). Thus it is possible for an X25 network to use the "More Data" flag to assemble small packets into larger ones inside the network, a feature which would promote packet efficiency and reduce the reassembly overhead at the destination. X75, however, only supports a maximum packet size of 128 octets, which is the size guaranteed by all X25 administrations. Intermediate reassembly is therefore only a reasonable option at the destination network, but by the same logic, fragmentation is only necessary in the source network (or the first STE encountered).

5.4 Flow Control Between Private and Public Data Networks

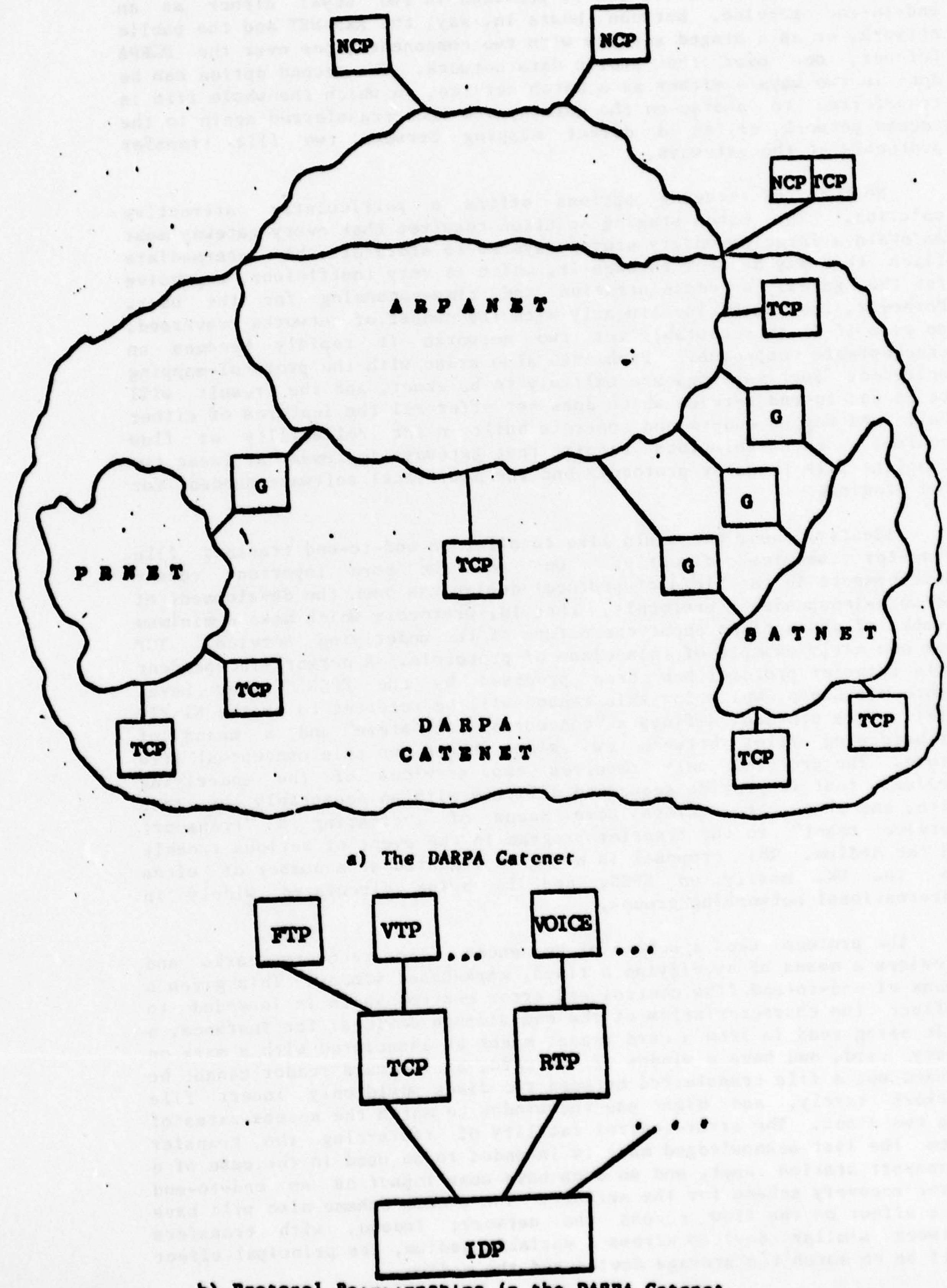
The connection of private data networks to public data networks is less well defined. The DTE which acts as a gateway between the two networks clearly supports X25 virtual circuits across the DTE/DCE interface, and this tends to encourage the extension of the concatenated virtual circuit model to cover the whole domain over which the end-to-end transport protocol operates. A datagram structure between the source host and the gateway will produce all the additional difficulties on flow control that we have noted in Section 4. Moreover, if the network is connected to the public network at more than one point and it allows dynamically routed datagrams to enter the public net through any of them, then all the potential advantages which the virtual circuit structure of the public networks can provide are lost.

This argument applies regardless of the end-to-end context within which the transfer is taking place. If an end-to-end transport protocol is in operation it will perform the necessary end-to-end controls; if not, the transfer can be broken down into independent stages. In either case, a concatenated virtual circuit structure offers the best way for a private network to make use of all available facilities for maximising throughput across the international public network, despite the costs of maintaining connection data.

6. An Experimental Transnet File Transfer System

It is clear from the above discussion that many of the architectural issues related to flow control for transnet services are very open. In particular, there is a real need to study the integration of flow control techniques at various levels within the transnet system, both to extract the maximum possible efficiency from the system and to overcome potentially harmful interactions which can occur without such integration. The section describes an experimental framework being developed to do this.

University College London has been involved in the study of networks for a number of years, notably in connection with the ARPANET, SATNET, and the experimental UK Post Office Network EPSS [23]. Over the next two years, the UCL configuration will undergo some considerable changes, as the direct ARPANET link is replaced by a transnetwork link through SATNET using TCP, EPSS is phased out, and we increase our involvement with a number of X25-based packet-switched networks. A simplified version of the new configuration is given in Figure 4. The central component of the new configuration is an LSI-11 which acts as a transport-level protocol convertor between the DARPA Catenet on the one side and the X25-based network on the other.



a) The DARPA Catenet

b) Protocol Relationships in the DARPA Catenet

Figure 2: DARPA Catenet Architecture

File transfer service can be provided in two ways: either as an end-to-end service, between hosts in, say, the ARPANET and the public network, or as a staged service with two components, one over the DARPA Catenet, one over the public data network. The second option can be done in two ways - either as a batch service, in which the whole file is transferred to a disc on the gateway and then transferred again to the second network, or as a direct mapping between two file transfer protocols at the gateways.

Neither of staging options offers a particularly attractive solution. The batch staging solution requires that every gateway must maintain a large secondary storage system to store all the intermediate files that may be sent through it, which is very inefficient, expensive for the gateway's administration and time-consuming for the user. Moreover, its costs rise linearly with the number of networks traversed, so even if it is acceptable for two networks it rapidly becomes an unacceptable approach. Problems also arise with the protocol-mapping solution. Such mappings are unlikely to be exact, and the result will be an end-to-end service which does not offer all the features of either half, and has no end-to-end controls built in for reliability or flow control. Both solutions require that gateways implement at least two complete file transfer protocols and the additional software needed for the staging.

Ideally, then, one would like to offer an end-to-end transnet file transfer service directly. One of the more important recent developments in the field of protocol design has been the development of network-independent protocols, that is, protocols which make a minimum number of assumptions about the nature of the underlying service. TCP was one early example of this class of protocols. A network-independent file transfer protocol has been proposed by the EPSS Higher Level Protocols Group, which for this reason will be referred to as the NI FTP [17]. This protocol defines a "conceptual file store" and a means of transferring files between two sides based on this conceptual file store. The protocol only requires two services of the underlying medium: that it provide sequenced delivery with an acceptably low error rate, and that there exists some means of indicating a "transport service reset" to the transfer program in the event of serious trouble in the medium. This proposal is being implemented at a number of sites in the UK, mostly on EPSS, and is being circulated widely in international networking groups.

The protocol uses a scheme of sequenced file recovery marks and provides a means of specifying a fixed, mark-based window. This gives a means of end-to-end flow control and error control which is intended to reflect the characteristics of the end storage devices: for instance, a file being read in from a card reader might be associated with a mark on every card, and have a window of one card, as the card reader cannot be backed up; a file transferred between two discs would only insert file markers rarely, and might use the window to match the access rates of the two discs. The error-control facility of restarting the transfer from the last acknowledged mark is intended to be used in the case of a transport station reset, and so does have some impact as an end-to-end error recovery scheme for the network. The window scheme also will have some effect on the flow across the network; indeed, with transfers between similar devices across a variable medium, its principal effect will be to match the storage device and the medium.

The major difference between this file transfer configuration and the ones discussed in the previous sections is that here there is no end-to-end transport protocol. One can, however, define a complete path of concatenated virtual circuits, with final reliability control residing in the file transfer protocol itself. On the SATNET side of the protocol convertor end-to-end transport is provided by TCP. It is, of course, also possible to build the NI FTP above the standard ARPANET NCP, in which case another connection would be set up between a host running both TCP and NCP somewhere on the ARPANET and the terminating host. The transport protocol across the X25-based net is at present undecided, but when it is determined it will be incorporated into the protocol convertor residing above X25. The protocol layer structure of the protocol convertor is shown in Figure 4.b.

The final system bears a strong resemblance to the architectural models we have developed for other transnet systems, especially that of Figure`1.b. Here, by analogy, we are defining a 'meta-net' within an end-to-end context provided by the NI FTP, and the 'local networks' are the DARPA Catenet, the public data networks, that part of ARPANET restricted to NCP communication, etc - i.e. they are based on a division by transport protocol rather than by physical network structure. To carry the analogy further, the meta-gateways of this system are the hosts which run different transport protocols within the framework of the 'meta-net'. Within the meta-gateways, above the transport services, the central module supports the specific requirements of the end-to-end application protocol defining the domain of the 'meta-net'.

The functions of this central module are in many respects similar to those of ordinary gateways. For instance, it must support similar addressing and routing functions, and it must set up and close down meta-gateway to meta-gateway virtual circuits, and support the opening and closing of the end-to-end connection. Its primary function is to support high-bandwidth end-to-end throughput rates by maintaining sequencing, by supporting and matching appropriate local flow control and congestion control strategies, by using locally available options for high-bandwidth flow, by tailoring the size of the data units to be as efficient as possible for the next transport section, and by indicating a "transport service reset" to the file transfer processes in the event of network failure.

The techniques which might be used to do this are the primary object of the investigation. Some initial suggestions are made here to indicate the nature of possible solutions. End-to-end control will be based on the file transfer window itself. At least in the early stages of the investigation, it will be assumed that transfers are disc-to-disc; hopefully, it will be shown that it will be possible to relax this restriction and still maintain adequate end-to-end flow. Over the DARPA Catenet, large window techniques would be adopted, based probably on conservative strategies for buffering and back-off retransmission strategies for congestion control. Similar strategies, appropriately modified, may be adopted over X25. Packets coming in from the X25 side may well be reassembled into TCP packets which have a size related to the record size of the file; in the other direction, they will be fragmented into 128 octet units, which contain exactly 2 file subrecords. Over X25 a high throughput class will be selected, and the unity of record fragments will be indicated by use of the "More Data" facility. User options in the DARPA Catenet present a considerable problem: as we have seen, none of those currently proposed are without

c) Network Access Gateway

Figure 1: Fundamental Transnet Architectures

undesirable side effects. Probably, none will be adopted until the position is clarified. It is expected that the flexibility of TCP packet sizes, combined with the close relationship between NI FTP subrecord sizes and X25 packet sizes should lead to an optimal fragmentation/reassembly policy.

7. Conclusions

Bulk data transfer is a network service which demands specialised support from packet switched networks in order to give efficient service. When the transfer is taking place across more than one network, the problems are compounded by the heterogeneity of the communications medium thereby created. Neither of the transnetwork architectures examined in this paper give entirely adequate support. It is instructive to examine the reasons for this. In the DARPA Catenet, we have an end-to-end transport mechanism which would seem to be ideal for bulk data transfer. The problems we encountered arise because of the minimal assumptions made about the underlying medium. Again and again we found that the fact that we can only assume a basic datagram service in the Catenet interferes with the effectiveness with which we can support the end-to-end service. Most of the problems could be greatly simplified if a virtual call service existed within the Catenet framework. From the standpoint of technical efficiency, the case for a virtual circuit option in the DARPA Catenet is strong. The major obbstacle to its implementation is the difficulty of making it compatible with adaptive routing.

The source of the deficiencies in public networks for supporting file transfer is rather different. Here we have the basic virtual circuit structure required, and we have some service facilities which are adequate for our purpose. The main problems are due to a deficiency which has often been noticed in the X25 debate: the lack of a global architectural framework. This makes it difficult to judge the effects of actions across X25 and X75 interfaces, as their end-to-end effects are indeterminate. The current version of X75 shows some awareness of this criticism, but flow control is one area in which no clear answers have emerged.

The two global architectures examined here differ in many respects. The experiment outlined in section 6 is an attempt both to find flow control policies suitable for bulk transfer in each, and to examine techniques for marrying the two to provide an efficient end-to-end service.

8. Acknowledgements

The work described in this paper was supported by grants from the Science Research Council, the Ministry of Defence, and the Defence Advanced Research Projects Agency. I would like to thank the referees, Dr Vinton Cerf of DARPA, and the many members of the INDRA group who have commented on this paper for their helpful suggestions.

out of date by the time of the conference, but in essence it should be unchanged.

Figure 2 shows the basic model behind the DARPA Catenet [33]. Currently the Catenet consists principally of the three DARPA-related networks given above, and gateways have successfully been built between ARPANET and SATNET [20] and ARPANET and the PRNET [21]. Other networks will be added in the future. The networks within the Catenet are

References

[1] - Bolt, Beranek and Newman Inc.: Specification for the Interconnection of a Host and an IMP. BBN TR 1822, January 1976.

[2] - C. J. Bennett, A. J. Hinchley: Measurements of the Transmission Control Protocol. Proc. Computer Network Protocols Symposium, Liege, February 1978 pG1-1. Also appearing in Computer Networks.

[3] - J. Burchfiel, W. W. Plummer, R. S. Tomlinson: Proposed Revisions to the TCP. INWG Protocol Note 44, September 1976.

[4] - CCITT: Recommendation X25. CCITT Orange Book, July 1977.

[5] - CCITT: Draft Recommendation X75. CCITT July 1978.

[6] - V. G. Cerf, R. A. Kahn: A Protocol for Packet Network Intercommunication. IEEE Trans. Comms., May 1974 p637.

[7] - V. G. Cerf: Gateways and Network Interfaces. IEN 6, April 1977.

[8] - V. G. Cerf, P. T. Kirstein: Network Interconnection. IEEE Trans. Comms., November 1978.

[9] - V. G. Cerf: The Catenet Model for Internetworking. IEN 48, April 1978.

[10] - K. M. Chuang: Private communication. August 1978.

[11] - S. W. Edge, A. J. Hinchley: Buffer Management Studies for Communications Network Host Protocols. INDRA 611, March 1977.

[12] - S. W. Edge, A. J. Hinchley: A Survey of End-to-end Retransmission Techniques. SIGCOMM, October 1978, p1.

[13] - S. W. Edge: Comparison of the Hop-by-hop and Endpoint Approaches to Network Interconnection. Proc. International Symposium on Flow Control in Computer Networks, Paris, February 1979 (this conference).

[14] - L. Garlick, J. Postel, R. Rom: Issues in Reliable Host-to-host Protocols. Proc. 5th Data Comms. Conf., Snowbird, September 1977, p4-58.

[15] - M. Gerla, G. de Stasio: Integration of Packet and Circuit Transport Protocols in the TRAN Data Network. Proc. Computer Network Protocols Symposium, Liege, February 1978, pB3-1.

[16] - G. R. Grossman, A. J. Hinchley: Issues in International Public Data Networking. Proc. USA/Japan Computer Conference, San Francisco, September 1978, p1.

[17] - EPSS Higher Level Protocol Group: A Network Independent File Transfer Protocol. Available as INWG Protocol Note 86, December 1977.

[18] - A. Hopper: Data Ring at the Computer Laboratory, University of Cambridge. Proc. NBS Workshop on Local Area Networks, August 1977, p11.

[19] - INWG: Proposal for an Internetwork End-to-end Transport Protocol. Available in Proc. Computer Network Protocols Symposium, Liege, February 1978, pH-5.

[20] - I. M. Jacobs, R. Binder, E. V. Hoversten: General Purpose Packet Satellite Networks. IEEE Trans. Comms., November 1978.

[21] - R. A. Kahn, J. Burchfiel, S. Gronemeyer, R. Kunzelman: Advances in Packet Radio Technology. IEEE Trans. Comms., November 1978.

[22] - P. T. Kirstein, M. Galland, D. Lloyd: Alternative Approaches to the Connection of Computer Networks. Proc. European Computing Conference, London, September 1975, p499.

[23] - P. T. Kirstein: University College London ARPANET Project Annual Report for 1977. UCL TR 48, April 1978.

[24] - L. Kleinrock, W. Naylor, H. Opderbeck: A Study of Line Overhead in ARPANET. CACM, January 1976.

b) Protocol Structure of a File Transfer 'Meta-Gateway'

Figure 4: An Experimental Configuration for Transnet File Transfer

[25] - R. M. Metcalfe: Packet Communication. Project MAC TR 114, December 1973.

[26] - J. Petersen: Remarks on the Implementation of the Packet Level Protocols of Public Packet Switching Networks. Proc. Computer Network Protocols Symposium, Liege, February 1978, pA2-1.

[27] - J. B. Postel: Specification of Internetwork Transmission Control Protocol, Version 4. IEN 55, September 1978.

[28] - J. B. Postel: Internet Protocol Specification, Version 4. IEN 54, September 1978.

[29] - A. M. Rybczynski, D. F. Weir: Datapac X25 Service Characteristics. Proc. 5th Data Comms. Conf., Snowbird, September 1977, p4-50.

[30] - J. F. Shoch: Internet Fragmentation and the TCP. IEN 20, January 1978.

[31] - V. M. Strazisar, R. Perlman: Gateway Routing, An Implementation Specification. IEN 30, April 1978.

[32] - C. A. Sunshine: Interprocess Communication Protocols for Computer Networks. SU-DSL TR 105, December 1975.

[33] - D. Walden, R. Rettburg: Gateway Design for Computer Network Interconnection. Proc. European Computing Conference, London, September 1975, p113.

NOTE: IENs are notes produced by the DARPA-sponsored Internet Group. Some of these are of limited circulation. INDRA notes are internal notes of the UCL INDRA group. INWG notes are produced by TC6.1 of the International Federation for Information Processing.

it generated the message. The "optimistic" policy bases window size calculations on some predictive heuristic to give a reasonable estimate of what the receiving TCP expects to possess in the near future. As one would expect, simulation studies [11] have shown that the conservative policy is the best one to follow, especially when buffer space management is the responsibility of the user process, which is the policy encouraged by the TCP implementors. In this case, an optimistic policy must try to gauge the demands that the user process is going to make on the TCP and the resources that it will make available. The

4.5 The EPSS-ARPANET-Satnet Demonstrations

In Fig. 2.1 we described the UCL INDRA configuration. We have two lines, at 48 and 2.4 Kbps, into EPSS. The EPSS is a Virtual Call network, with a well-defined set of protocols for Network Access, Transport Station, Virtual Terminals and File Transfer. None of the protocols are very similar to the ARPANET ones, though reasonable mappings are possible, e.g. between TELNET on ARPANET and a Virtual Terminal Protocol (VTP) on EPSS. UCL has set up a mapping gateway, so that Terminal streams over EPSS can be sent over ARPANET. This system is described in some detail elsewhere (23).

In the documentation on the INDRA software systems Gnome (24) and Santa (25), we showed how terminals attached to an ARPANET TIP at UCL could be used to control Data Generators sited in the ARPANET-SATNET gateways. A TENEX on ARPANET was used to control the experiments; communication with the Controller was via terminal interaction over ARPANET.

In June 1978, a large demonstration was organised to show the capabilities of EPSS. As part of this demonstration, the capabilities of the two previous paragraphs were combined to allow Satnet measurements to be performed by a concatenation of SATNET, ARPANET and EPSS. The forms of connection used were chosen for pragmatic reasons; they required extensive protocol conversions at several points. Thus, the demonstrations do not present a reasoned approach to how Internet Services should be provided in the future. However, a very effective Internet demonstration using all three networks was set up, which worked well.

The physical topology of the EPSS-ARPANET-Satnet demonstration is shown in Fig. 4.1. The relevant portions of ARPANET are the IMPs at BBN and ISI to which were attached the Satnet Monitor Center TENEX (SMC) and the TENEX (EC) running the Santa Controller. While the TENEX or TOPS 20 running the Data Acquisition software could be a different machine, we usually used EC also for this function. To it Norsar, BBN and UCL TIPs (or IMPs) were attached to PDP 11/34 Gateways. These Gateways were Hosts on both ARPANET and EPSS. For convenience, character terminals were attached to the UCL TIP.

For data processing between ARPANET and EPSS, the UCL PDP9 did protocol mapping of the End-to-End Transport Protocols and the Virtual Terminal Protocols. The mapping was based on concatenated virtual calls. No Internet functions (like the Internet packet format) were used. The data passing between the Experiment Controller Santa and the Gnome traffic generators in the Gateways were raw ARPANET packets. The Gnome traffic generators produced packets with the correct internet leaders to pass through the ARPANET-Satnet gateways to their data Acquisition counterparts in the destination gateway(s).

Character terminal traffic originated both from terminals attached to the UCL TIP, and from terminals directly dialling into EPSS. Since all UCL character terminal multiplexing is performed by the UCL TIP, the communication path between the UCL terminals, the Santa Controller and the Satnet Monitoring Centre were complex. All terminal traffic passed from the TIP (T in Fig. 4.1) with protocol conversion via P2 and EPSS over the 2.4 Kbps line. Both the traffic from the UCL terminals and the others then passed through EPSS and the 48 Kbps line through P1 (with EPSS-ARPANET protocol conversion) to the TIP. From there a normal ARPANET connection was made to the EC and SMC; the terminal dialog between UCL and EC resulted in control traffic going between EC and Gnome in the controlling Gateways CG.

This traffic itself generated control information between Gnome in GC and that in other gateways G through Satnet. The Controllers in the G generated experimental traffic over Satnet which produced statistics data in the Gnomes in G. This statistics data was passed through ARPANET to the Experimental Acquisition system (EA usually the same as EC). Finally, the statistics were partially preprocessed in EA, and displayed on the UCL terminals via ARPANET. They were also sent via the UCL TIP and P1 through ARPANET and EPSS with the appropriate protocol conversion, to the RL. Here an IBM 360/195 analysed fully the data and produced output on an electron beam recorder. This whole process, while convoluted, worked reliably - and demonstrated the multi-net capability already in existence.

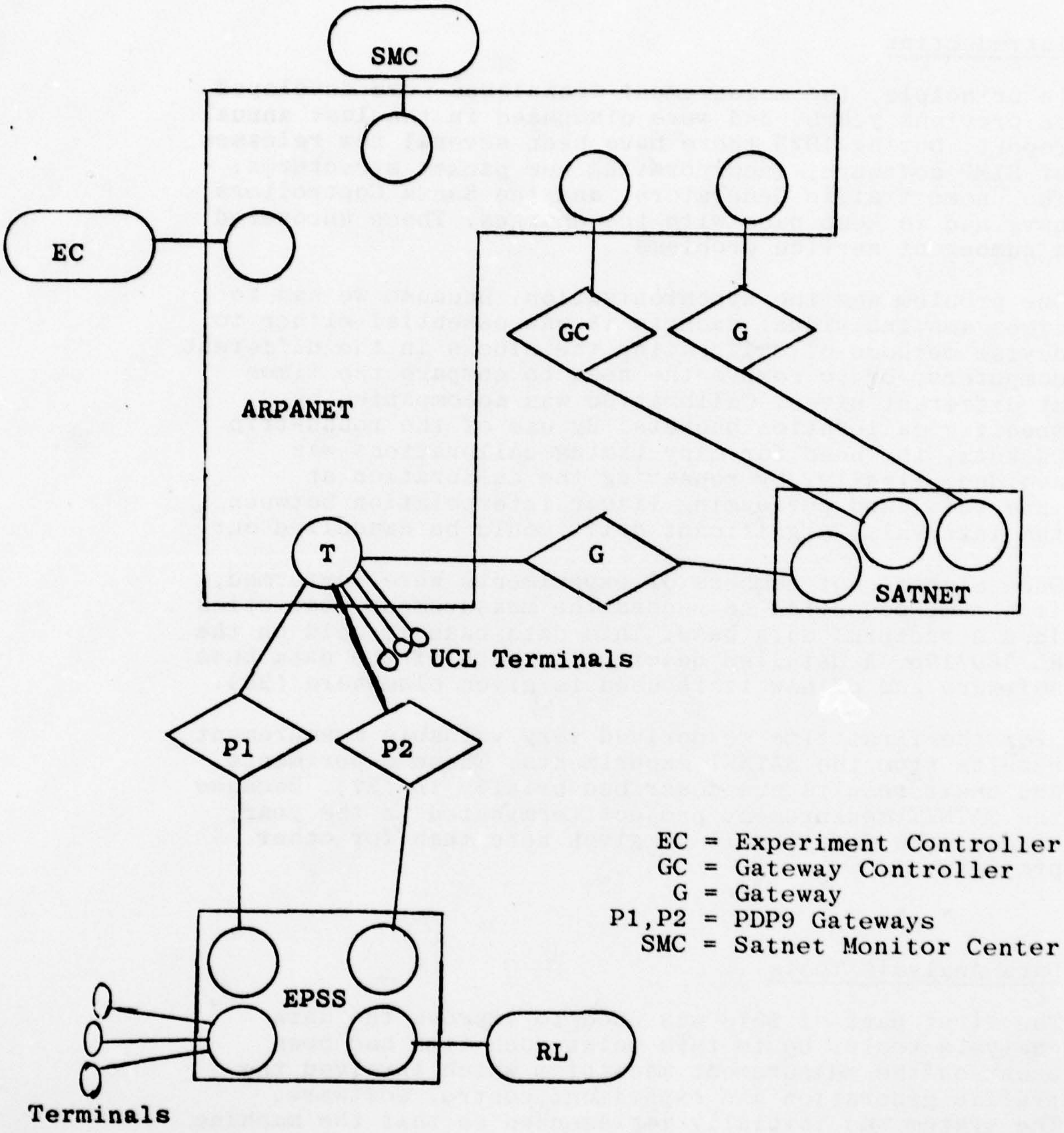


Fig 4.1 Schematic of Coupled Networks

V. SATNET ACTIVITY

5.1 Introduction

In principle, the measurement techniques were developed in previous years, and were discussed in the last annual report. During 1978 there have been several new releases of SIMP software, incorporating new packet structures. The Gnome traffic Generators, and the Santa Controllers, have had to keep pace with the changes. These uncovered a number of service problems.

One problem was the synchronisation. Because we had to timestamp individual packets it was essential either to devise methods of calibrating the clocks in the different computers, or to remove the need to compare the times at different sites. Calibration was accompanied by specific calibration packets. By use of the round-trip packets, the need for many timing calibrations was avoided. Finally, by repeating the calibration at intervals, and performing linear interpolation between the intervals, significant drift could be cancelled out.

Once significant numbers of experiments were performed, it became essential to record the measurement statistics in a structural data base. This data base is held on the RL 360/195. A detailed description both of the data base software and of how it is used is given elsewhere (26).

For the first time we derived very valuable measurement results from the SATNET experiments. These experiments and their results are described briefly in (27). Because the SATNET measurement project terminated in the year, rather more detail will be given here than for other projects.

5.2 Data Analysis Tools

The first part of 1978 was used to improve the data analysis tools. Up to this point much time had been spent on the measurement mechanism which involved the traffic generation and experiment control software. The system was initially implemented so that the machine that controlled the experiment also collected the data. As the controlling machine was at ISI (Los Angeles) in the US, the data at the end of the experiment was collected there too.

Even though the data was collected in the US, a decision was taken to perform all the analysis on the IBM 370/195 at Rutherford Laboratory in Oxfordshire. Rutherford was selected as it not only had plenty of computer power but also had an impressive set of graphics hardware. The main graphics machine was the FR80 electron beam recorder, a machine that enabled many different forms of graphics output to be produced, from large black on white paper hardcopy to the production of 35mm colour slides. We found the ability to produce slides of our results an extremely important asset.

At the end of each experiment, the data was transferred from the ARPANET where it had been collected to the Rutherford Lab via the London Gateway at University College. This operation was usually completed within 30 minutes of the end of an experiment, although the actual time depended upon the amount of data to be transferred. (The data could be transferred at the rate of 1000 bits per second).

At Rutherford the data was analysed by a suite of programs and placed on a database that was designed and created especially for the purpose of collecting these statistics. The database had the ability to collect approximately 30 experiments online at a time. After the database had been filled the older experiments could be archived on to magnetic tape. Such archived experiments could be rapidly restored onto the database if the need arises.

The graphics programs were written to access this database and extract experimental information. These routines would collect data, according to user-specified parameters, and display the results in graphics form. These routines had the ability to display a crude drawing on the line printer or to produce rather better graphics on the Computek display unit that we have on site in London. The experimenter could then route the graphics output to the FR80 and the colour slides would be received about a week later. The database access routines also had the ability to compare the results of different experiments, and even to use the information of a complete set of experiments to deduce trends in the data.

5.3 Experiment Runs

The actual experiments began early in the year, but at first these were just trial experiments to prove that that system was working correctly.

Initially there were only three types of traffic generation possible: Stream Traffic, Bulk Traffic and Interactive Traffic. Stream Traffic was found to be important for the measurement of the network capabilities as this method enabled messages to be generated at fixed time intervals. As the inter-message time at the transmit side was constant the inter-message time at the destination could be measured to see how it differed from the source side. The disturbance introduced into the traffic stream could then be measured directly. As the frequency of packets was increased it was also possible to see how the network reacted to the heavier traffic load.

The Bulk Traffic mode was changed shortly after the first set of experiments to enable it to simulate a windowing technique. This transmission method meant that there was a "window" of messages at the source such that only this window size was permitted to be outstanding in the network at any one time. Transmission at the source is then frozen until an acknowledgement is received for at least part of the data. This technique was based heavily upon TCP concepts and used parts of the same algorithm.

It is interesting to note that the interactive traffic type was never found to be very useful. The interactive traffic was tested on a few occasions but the results were found to be difficult to analyse due to the number of variables involved. The stream traffic type and the bulk traffic type were found to achieve far more in terms of network measurement. The interactive traffic was based upon messages being sent back from the destination each time a message was found that the response of the network could be measured more accurately with the stream traffic type.

5.4 Measurement Limitations

Unfortunately the generation machine was found to be too slow to explore fully the data capacities of SATNET. The machines could not generate traffic faster than 8 packets per second, far below what the network could be expected to accept. This was attributed to the operating system which although very good from a debugging point of view was very slow in terms of processing packets from a user process to the network or vice versa. Eventually we found that we had to live with this limit of eight packets per second and concentrate instead on the disturbance and delays occurring in this traffic rather than pushing a high data volume through the network.

Our techniques permitted the detection of at least one "bug" in the network software. We found that even at low data rates the packets would arrive wildly out of sequence at the destination. This was later found to be a bug in the satellite IMPs whereby the test to place the packets in the output queue had been reversed so that the packets were placed at the front of the queue when they should have been at the end.

5.5 Measurement Results

The experiments were performed in several phases. The first set of experiments consisted of echoing traffic from different points in SATNET. This echo loop was started as an internal loop within the source machine and then expanded until it looped traffic to the local SIMP, then to the satellite, the destination SIMP and finally the destination machine. These experiments allowed the behaviour of the network to be studied in some detail and the bottlenecks of the system to be determined.

Essentially, we found that the bottleneck occurred in the source machine. The maximum packet rate internally looped within this machine was just over 12 messages per second. (Compare this to another ARPANET experiment that had a microprocessor handling 200 messages per second). When the loop was extended to the SIMP so that the traffic travelled along the VDH line and back again the throughput dropped to 10 messages per second. Thus the extra overhead of transmitting the packets to the SIMP and back reduced the throughput only slightly.

For most traffic across the network we measured a delay of between 1.1 and 1.75 seconds, with a peak at 1.4 seconds. This was measured with packets being sent every 140milliseconds, but when this frequency was increased the delays went up dramatically, mainly due to the queueing in the source machine being unable to handle them.

The windowing experiments achieved some very interesting results. Here we measured the time from sending a packet to the time a reply (acknowlegement) was received and found that this varied with the size of the window. The best round-trip time of 2 to 3 seconds was achieved with a window size of 6 messages: when the window was expanded to 15 messages the maximum delay rose to 4 seconds for the round-trip. In fact once the transmission rate at the source exceeded the throughput that the network could take the message delays increased until eventually all the buffer space was exhausted. Thus we saw that opening up the window larger than the throughput rate of the network not only does not improve the throughput but is detrimental to the other users of the network. Once the window size gets too large the messages sit in buffers waiting for transmission and thus prevent other tasks from using the buffer resource.

5.6 Conclusions

The measurement results more than justified the time spent in the development of the measurement tools. The system we developed was able to measure many aspects of computer network performance. The performace of individual machines could also be determined. Even more impressive was the fact that these measurements could be run and co-ordinated in machines that were many thousands of miles apart in different continents. Within minutes of an experiment terminating we could have the first results displayed on our screens. This rapid access to results and the ease with which complex experiments could be run enabled us to pursue an extensive set of experiments in a relatively short time.

Perhaps even more important than the results themselves was the experience we gained in timestamping, the foundation of our measurement effort. After several months of measurements we built up an enormous amount of experience in the use of timestamps. We had to solve various problems associated not only with the mechanics of collecting timestamps but also of varying clock rates between machines operating at great distances. We approached the problem of global clock synchronisation and attempted to find solutions. We then moved on to develop measurement methods that did not require clocks to be synchronised, and yet the timestamps were still able to yield detailed measurement results.

VI. FACSIMILE ACTIVITIES

The actual grant under which our facsimile work has been carried out expired in July. A final report on that project has been written (28).

The previous annual report included two papers which summarised much of our work. These papers were somewhat revised before publication (29,30). Another paper mentioned in the previous report, on terminals for facsimile has also appeared since that date (31). These papers are only mentioned because the previous reference list in (1) was incomplete. A much more significant extension and revision, with in-depth detail, is presented in Yilmaz's Ph.D. thesis.(32)

During the first half of 1978, we tried to add X25 software and hardware to our Intel 8080 facsimile terminal. In the end both activities only had useful training value. Although the exercise was successful, the resultant software was so large, that it was impractical to have it coexist with our facsimile software. Our hardware was purchased as a Microcomputer in 1975, and could not support multiprocessors. Others (e.g. the European Informatics Network Executive) have also found that a multiprocessor architecture is more appropriate with this generation of microprocessor for X25 based applications.

A Digital Facsimile device, a Dacom 450, was obtained during the year. This device has both a built-in modem and a V24 modem interface. The block structure used in that machine (33) is such that it can be used by running the HDLC interface built for our LSI-11s in a transparent way. The interface has to be run in a strange block synchronous mode; it has to synchronise in a 6-bit mode and then switch dynamically to 8-bit in a data phase. However, it has proved quite feasible to control the device from a local LSI-11. By interrupting the Clock signal, it is even possible to halt the machines for flow control purposes - which was not feasible on the earlier 6K machines.

We have defined a simple set of procedures for storing and retrieving facsimile files. These procedures fit above a standard transport level of protocol. The system is illustrated in Fig. 6.1.

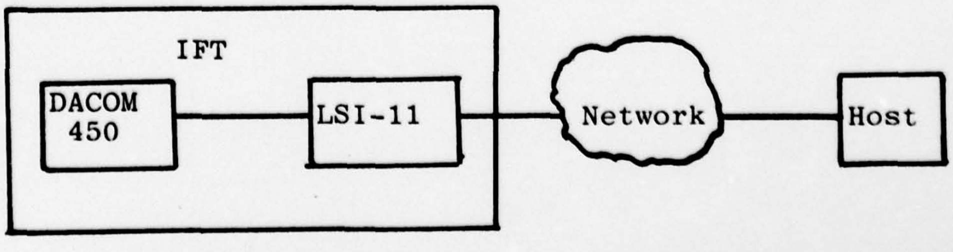


Fig. 6.1 Schematic of Terminal Facsimile Service

IFT = Intelligent Facsimile Terminal.

We are building the software in a modular form so that the network interfaces to IFT can be either a standard ARPANET 1822 or a standard HDLC one. The network access and transport software will be the ARPA Internet TCP for the first case, and X25 plus transport level for the second. Thus we hope to develop a genuinely versatile terminal. We have started developing standard software for filing and retrieving single papers for three computers above the TCP level; these are the DEC TOPS 20, the PDP-11 with UNIX, and the PDP-11 with the MOS operating systems.

Based on these building blocks, we hope to be able to provide a prototype service over ARPANET /Satnet by the summer of 1979, and over Euronet by late 1979.

A new facsimile-related project is starting in 1979, with the sim of integrating facsimile with other office automation functions.

VII. SIMULATION ACTIVITIES

7.1 Introduction

Earlier annual reports have mentioned the simulation activities of the group. However, they were not in a form to be described easily in our annual report. Now two publications have been produced on this work. These are reproduced as Sections 7.2 and 7.3. In the first we compare the relative advantages of Hop-by-Hop and end-to-end procedures - using extensive simulation techniques. In the second more detailed computations were used to contrast different end-to-end transmission strategies for the TCP protocol.

7.2 Comparison of the Hop-by-Hop and Endpoint Approaches to Network Interconnection

COMPARISON OF THE HOP-BY-HOP AND ENDPOINT APPROACHES TO NETWORK INTERCONNECTION

Stephen William Edge

Department of Statistics and Computer Science

University College London

This paper compares two principle architectural approaches to network interconnection. These are the Endpoint approach, which uses a standard internet end-to-end protocol, and where network support can be minimal; and the Hop-by-Hop approach, in which internet calls consist of a sequence of local virtual calls across each intermediate net. These approaches are first contrasted very generally and certain situations are identified which heavily favour just one of them. Concentration specifically on flow control follows. A major point to emerge is that an Endpoint architecture is likely to demand more resources from Hosts; whereas a Hop-by-Hop Architecture demands more from Gateways and may produce inferior service. Such differences are illustrated, for a particular internet connection, by simulation and analysis.

1. INTRODUCTION

Interconnection of packet switching networks has occured only recently, and to a limited extent. It is intended to provide a wider selection of user services, by creating in effect an enlarged net - which will be referred to here as a "Catenet" (Pouzin 74). In this, individual nets are joined at their boundaries by what have come to be known as "Gateways" (Cerf 74, Pouzin 74, Sunshine 77a). One example of an operational Catenet is the "TCP Catenet" (Cerf 78c, Bennett 79) linking the ARPANET (Roberts 73) to a number of private datagram nets, including SATNET (Jacobs 78), the ARPA Packet Radio Net (Kunzelman 78), and the Xerox PARC Ethernets (Metcalfe 76). Another example is the EIN Catenet linking the EIN, CYCLADES and NPL nets (Deparis 76).

A Gateway can be regarded - rather generally - as an entity - or entities - interfacing connected nets, and residence within such nets is not precluded (Sunshine 77a). There is general agreement that Gateways, which reside outside the nets they interconnect, should interface to these nets as Hosts, rather than say as Packet Switches (Lloyd 75, Sunshine 75, Sunshine 77a). This has lead to a standard model of a Gateway as a single physical entity, attached to two or more networks as a Host (Pouzin 74, Binder 75, Gien 75, Lloyd 75, Walden 75). The logical Host portions in such a Gateway are sometimes referred to as "Gateway Halves" (Sunshine 77a), which we shall abbreviate to "Gate Halves". Figure 1 illustrates such network interconnection, and we shall assume this model in the rest of this paper.

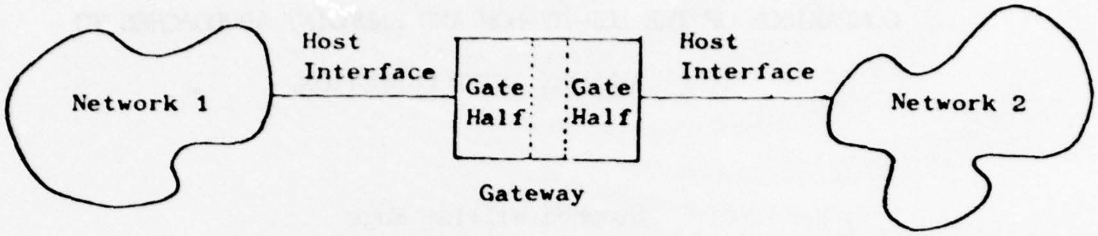


Figure 1. Interconnection of Networks through a Gateway

Opinion is divided on the protocol level at which traffic, on individual internet end-to-end connections should be transferred through Gateways, between the Gate Half portions. The two main alternatives are transfer at the datagram level using a standard internet datagram format (Pouzin 74, Gien 75), or transfer at the Virtual Call Level (Gien 75, Lloyd 75, Sunshine 77a). Transfer at an even higher level - e.g. the Virtual Terminal Level - is also possible (Higginson 75). These alternatives lead to two distinct architectural approaches to network interconnection, which have been labelled "Endpoint" and"Hop-by-Hop" (Sunshine 77a).

In the Endpoint approach - illustrated in Figure 2 - internet traffic transfer through Gateways is at the datagram level, and Hosts must use a standard internet end-to-end protocol - such as TCP (Cerf 78a) or INWG 96 (Cerf 78b). Transfer of internet traffic between adjacent Gateways and Hosts, may then use minimal local net services. In the Hop-by-Hop approach, illustrated in Figure 3, no standard end-to-end protocol is required; internet traffic transfer through Gateways is at the Virtual Call level. Here internet connections are composed of a chain of local virtual calls across intermediate networks, and Gateways may need to translate between different virtual call protocols. This approach also permits Application Level protocols (e.g. Virtual Terminal Protocol) to be translated at Gateways (from one net to another) if there is no internet standard.

This paper is concerned with a comparison of these approaches, particularly with regard to flow control. We begin (Sections 2 and 3) by comparing the two approaches very generally, and show that some situations heavily favour just one of them. Concentration specifically on flow control follows (Section 4) and is deemed most relevant to situations in which either approach is feasible. A major point to emerge is that an Endpoint architecture is likely to demand greater resources from Hosts, whereas a Hop-by-Hop Architecture demands more from Gateways and may produce inferior service. Such differences are illustrated for a particular internet connection by simulation and analysis (Sections 5 and 6).

2. ENDPOINT INTERCONNECTION

Endpoint interconnection is used in both the TCP Catenet - with TCP as the internet end-to-end protocol - and in the EIN Catenet, which uses the EIN end-to-end protocol (Deparis 76). Its main advantage is in enabling simple Gateways. These are not required to handle individual virtual calls, and may use local net datagram services - where possible - to transfer internet packets.

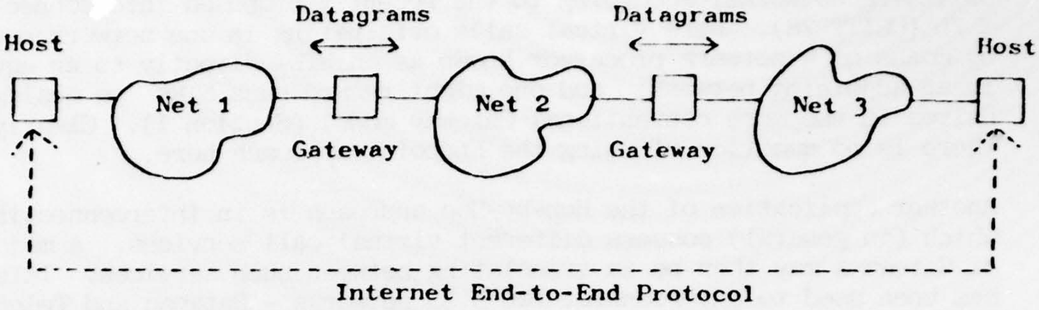


Figure 2. The Endpoint Interconnection Approach

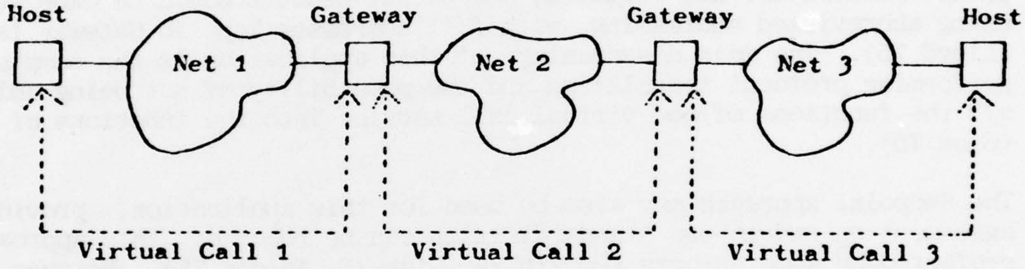


Figure 3. The Hop-by-Hop Interconnection Approach

Another advantage is that inter-Gateway routing can be dynamically altered, on a packet-by-packet basis, allowing rapid response to network failures and congestion.

The main disadvantage is the requirement of end-to-end standardisation. Local nets are likely to have implemented their own Host level protocols. The addition of an internet standard will not only be costly to implement, it may also leave some "disadvantaged" Hosts, unable to engage in internet communication. This has occured in both the TCP and EIN Catenets (Kirstein 78, Deparis 76), and has required the construction of special Hosts - known as "internet service sites" (Sunshine 77a) - which implement the internet protocol on behalf of those disadvantaged Hosts, in their own nets. Other disadvantages of this approach include the likelihood of large packet headers, which would cut into usable communication bandwidth (Postel 78), and the unnecessary duplication by Gateways of end-to-end features, when internet traffic passes over a virtual circuit net.

The Endpoint approach is favoured mainly for interconnecting datagram networks - such as in the EIN and TCP Catenets - where there is agreement on end-to-end standardisation.

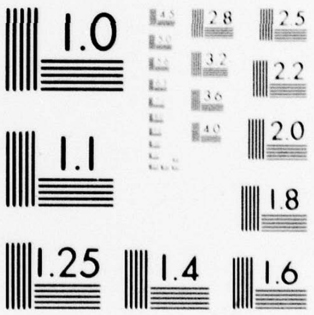
3. HOP-BY-HOP INTERCONNECTION

There are two distinct applications of the Hop-by-Hop approach. The first,

AD-A074 489 UNIVERSITY COLL LONDON (ENGLAND) DEPT OF STATISTICS --ETC F/G 17/2
UNIVERSITY COLLEGE LONDON INDRA PROJECT.(U)
MAR 79 P T KIRSTEIN N00014-77-G-0005
UNCLASSIFIED TR-56 NL

2 OF 2
AD
A074489

END
DATE
FILMED
11-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

which we mention only briefly, involves interconnection of public packet switching networks, according to the recently proposed interconnection standard X.75 (CCITT 78). Here virtual calls originating in one network are relayed - by means of a network processor known as an STE- directly to an equivalent STE in an adjoining network; and one might regard such STEs as analagous to Gate Halves in the more conventional Gateway model (Section 1). Clearly, however, there is no question of using the Endpoint approach here.

Another application of the Hop-by-Hop approach is in interconnecting networks, which (in general) possess different virtual call services. A major function of Gateways may then be in translating between such services. This approach has been used to interconnect two X.25 networks - Datapac and Telenet (Rybczynski 78), and to interconnect ARPANET and EPSS (Higginson 78). We proceed to discuss this application in more detail.

The main advantage of the above Hop-by-Hop application is to allow the use of existing virtual call services for making internet connections. Another advantage is that unlike the Endpoint approach, long internet end-to-end packet headers are not required, and packet headers could be made very short using abbreviated addressing, with full addresses kept in Gateway tables (Lloyd 75). The main disadvantage of this application is the complexity of performing protocol translation and the possibility of not being able to map all the functions of one virtual call service into the functions of another (Gien 75).

The Endpoint approach may also be used for this application, provided an agreement on end-to-end standardisation can be reached; this approach is preferred by some authors (Pouzin 74, Gien 75, Walden 75). However, as we have seen, both approaches possess advantages and disadvantages, and in a practical situation one would have to assess the relative importance of these as well as of other relevant criteria. In the rest of this paper, we concentrate on one such criterion, flow control, which has previously been neglected.

4. FLOW CONTROL CHARACTERISTICS OF NETWORK INTERCONNECTION

The objectives of flow control have been defined as the provision of adequate end-to-end service (throughput and delay) and the minimisation of the relevant network resources (storage, transmission bandwidth, and processing overhead) (Pouzin 76). With network interconnection, one interconnection approach is superior to another if it provides better service using the same resources - or equivalently needs fewer resources for the same service level. We proceed to evaluate the Endpoint and Hop-by-Hop (including X.75 Hop-by-Hop) approaches in this respect. Criteria of interest will comprise end-to-end delay, throughput limitation, and Host and Gateway resource requirements. Consideration is also given to the requirements of certain traffic types (e.g. bulk traffic).

We look first at the Endpoint approach. When used to interconnect datagram networks, service could be optimal, both because packets can be delayed minimally at Gateways and because the ability to use adaptive routing allows flexible response to internet and local net congestion. However, service would be sub-optimal whenever traffic had to cross a (e.g. public) virtual circuit net, due to unnecessary local net sequencing of internet traffic (Sunshine 77a).

Resource requirement in the Endpoint approach inside Hosts, both in terms of storage use and processing time, is likely to be larger than in the Hop-by-Hop approach. This is because end-to-end internet protocols, such as TCP and INWG 96, assume minimal subnet level service, and so require extensive and complex Host action. Further, such protocols require internal Host storage, both for packet retransmission (at a sender) and for message reassembly, sequencing and delivery (at a receiver). Such storage is related to the amount of outstanding (unacknowledged) traffic at a sender, and would be particularly large for bulk (or real time) data transfers over long internet connections. In contrast, the functions of Gateways can be less complex, and buffer storage for packet retransmission and sequencing (if used at all by Gateways) would be required only over single nets, where the amount of outstanding traffic would be limited.

The main feature of the Hop-by-Hop approach, from the point of view of Hosts, is to make internet connections look like local net connections with respect to resource utilisation. Thus, Host storage is only required to support traffic over a single net, and would be less - particularly for bulk data transfer - than in the Endpoint approach. Further, local protocols could be simpler and less time-consuming to manage than a complex internet protocol, and it would be unnecessary to manage simultaneously two (local & internet) protocols.

Conversely, Gateways in the Hop-by-Hop approach must maintain individual virtual calls for each internet connection they handle. This implies greater processing and storage requirements than in the Endpoint approach. Furthermore, service in the Hop-by-Hop approach is likely to be sub-optimal, due both to unnecessary sequencing on each virtual call hop, and to extra delays associated with connection management and protocol translation. A necessity for protocol translation might also limit the throughput obtainable across a Gateway.

These observations imply several tradeoffs between the Endpoint and Hop-by-Hop approaches. The Endpoint approach is capable of better service, requires fewer Gateway resources, but requires more Host resources; whereas the Hop-by-Hop approach limits Host resource requirements to those for ordinary local net connections. We demonstrate these tradeoffs in the following sections, using a simulation model of an internet connection.

5. INTERNET SIMULATION MODEL

In this section we describe a simulation model of an internet end-to-end connection between two communicating processes. These are resident in Hosts, separated by "n" intermediate nets, and "n-1" Gateways - Figure 4 - and a fixed internet route is assumed. Each Gateway is composed of two Gate Halves, according to the model of Section 1. Two levels of protocol are included: an internet end-to-end level between the two Hosts, and a local net virtual call level between adjacent Gate Halves, which we shall refer to as the Gateway level. The latter protocol also operates between each Host and its local Gateway, and so we consider each Host to contain a logical Gate Half portion, as well as a logical end-to-end portion. Different mechanisms will be used at each of these protocol levels to enable modelling of both the Endpoint and Hop-by-Hop approaches.

5.1 PROTOCOL MECHANISMS

The protocol mechanisms, which can be used at each level, comprise the following:

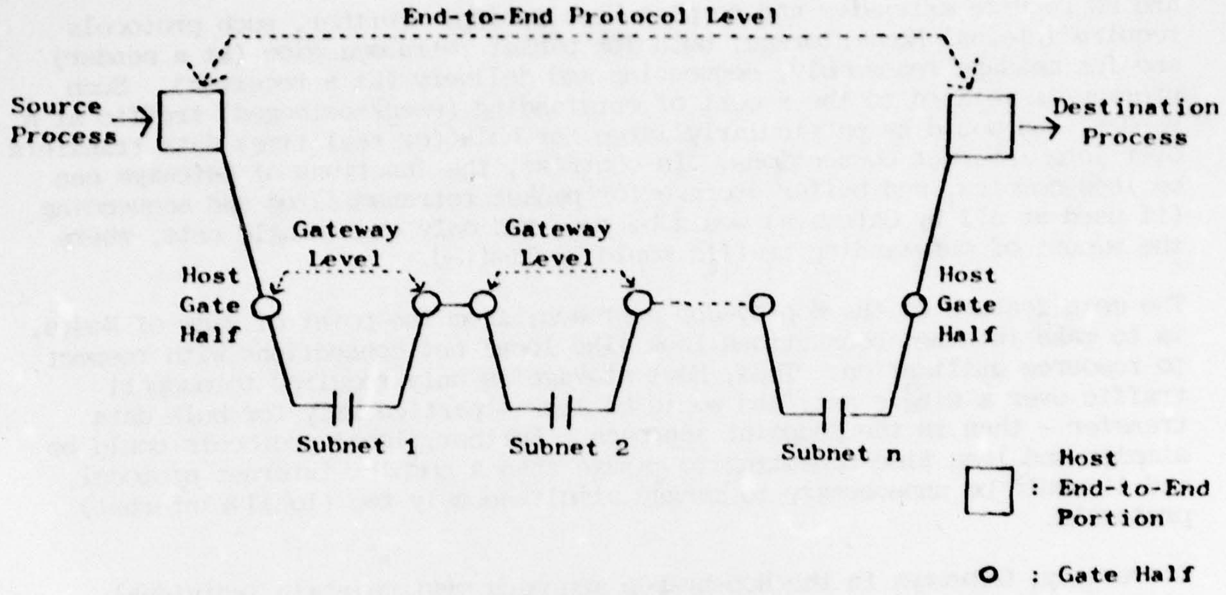


Figure 4. The Internet Simulation Model

Nil	:	No Protocol
PAR	:	Positive Acknowledgement and Retransmission
SPAR	:	Sequencing, Positive Acknowledgement and Retransmission

We shall denote by the term "Nil" the removal of either level of protocol.

The "PAR" protocol (Sunshine 75) operates between a Sender and a Receiver of packets. Each packet transmitted by the Sender is allocated a unique identifier, and a copy is retained for retransmission using internal buffer storage. The Receiver immediately acknowledges each correctly received packet, by returning its identifier in a special acknowledgement packet. Acknowledged packet copies are discarded at the Sender, and unacknowledged packets are retransmitted (repeatedly if necessary) after a timeout interval.

The "SPAR" protocol (Sunshine 75) is a more complex version of the "PAR" protocol. Transmitted packets are allocated a monotonic increasing sequence number, and packets arriving at the Receiver are re-ordered in sequence, using internal storage. Whenever possible, single packets or contiguous packet blocks are delivered in sequence by the Receiver to the next stage, and a single acknowledgement packet is returned, carrying the sequence of the next packet to be delivered. This acknowledges all preceding packet deliveries. The Receiver also recognises and discards duplicate packet arrivals (for packets already delivered or being re-ordered) and a positive acknowledgement is returned to the Sender, re-acknowledging previous deliveries. The retransmission procedure at the Sender is identical to that in the PAR protocol.

The PAR protocol is intended mainly for efficiency, to re-supply lost packets, and it can introduce packet duplicates. Such a protocol is used, for example, between packet switches in the ARPANET (McQuillan 77). The SPAR protocol is the basis of a Host level virtual call protocol, since it delivers packets in sequence, without loss or duplication - in the absence of system crashes and provided certain constraints on the re-use of sequence numbers are obeyed (Sunshine 75). We shall use it to represent both an internet end-to-end protocol, and a Gateway level local virtual call protocol. Because comparisons of different approaches are made at a constant throughput (i.e. constant service) level, additional mechanisms for flow constraint - e.g. a window (Cerf 74) - are not used at either protocol level.

We shall denote the protocol mix used in the simulation model by the convention A(B1,B2, ..., Bn), with "A" denoting the end-to-end mechanism and Bi denoting the Gateway level mechanism across the i'th net.

5.2 MODEL DESCRIPTION

Data flow in the model is one way from a "Source Process" and "Source Host" to a "Destination Host" and "Destination Process". The Gateway level protocol supports both forward data transfer and the return of end-to-end acknowledgements. We assume the two protocol levels act independently, with no piggybacking of Gateway level acknowledgements on end-to-end packets.

We describe the model in terms of its three constituent levels: end-to-end, Gateway-to-Gateway, and subnet - Figure 4. We assume that (at Gateways or Hosts) packets are transferred between these levels instantaneously, and we are not concerned with blocking or processing delays by any level. In the following description, the end-to-end and Gateway levels are assumed implicitly to implement one of the protocols described in Section 5.1.

At the end-to-end level, the input is a steady stream of messages arriving at the Source Host from the Source Process, with constant inter-message arrival time A. Each message is encapsulated in a single packet and passed (immediately) to the next level (the Host Gate Half portion). At the Destination Host, the messages in packets handed over by the Host Gate Half portion, are passed - possibly after sequencing - to the Destination Process (assumed always able to accept them), and a positive acknowledgement may be handed back to the Host Gate Half for reverse transmission.

The Gate Halves of the Gateway level take their input from, and send their output to, either an adjoining Host end-to-end portion (in the Hosts), or an adjoining Gate Half (in the Gateways). Gate Halves are assumed not to fragment packets. Packet transfer between opposite Gate Halves is through the subnet level. This imposes a variable delay and the possibility of loss on each packet transferred. An Erlang distribution is used to model the delay of each subnet. This has a probability density function f, which, for a delay x, is given by:

$$f(x) \; = \; (k.u)^{k}.x^{k-1}.e^{-kux}/(k-1)! \qquad x>=0 \qquad (1)$$

The parameter 1/u is the mean delay, and $k^{-\frac{1}{2}}$ is the delay coefficient of variation. When k is 1, f(x) is the exponential distribution. Larger k produces a distribution with a sharp peak near the mean and with a long tail extending to x equals infinity. This corresponds to the types of delay encountered in practice (Kleinrock 77, Gien 78). We shall set k to 25 in all cases, representing low delay variation (the corresponding delay coefficient of variation is .2). Loss of packets in transit, or loss due to insufficient buffer space at Gateways, is represented by applying a fixed probability of loss p independently to each packet in transit. The delay and loss in each subnet are identical for packets and acknowledgments travelling in either direction. We note that for high packet loss, the exact value of k, used for eq'1, will be less significant, since retransmission (following packet loss) will then be a major cause of long packet delay.

The following summarises the parameters used in the model:

A : Inter-arrival time of messages at the Source Host
R : Source Host end-to-end retransmission timeout
R_i : Gateway level retransmission timeout across the i'th net
k_i : value of k in eq'l for the i'th net
u_i : value of u in eq'l for the i'th net
p_i : packet loss probability in the i'th net

Performance measures for the model comprise the end-to-end message delay, the retransmission overhead at each protocol level, and Gateway and Host storage requirements. Appendices 1 and 2 present an analytical derivation of certain of these, under various protocol mixes, and this allows partial validation of simulation results.

6. SIMULATION RESULTS

The conclusions of Section 4 are illustrated with simulation results, using a GPSS program, for the internet connection model of Section 5. Where possible, analytical results (Appendices 1 and 2) are used to validate corresponding simulation results. When these agree closely, only the (more accurate) analytical results are shown in the graphs below. Otherwise, both sets of results are shown. Analytical results are identified with an asterisk.

We look at an internet connection spanning two identical nets, and carrying bulk traffic, for which resource requirements are significant (Section 4). The following parameter settings are used, with the average end-to-end subnet delay ($1/u_1 + 1/u_2$) set to one unit of time:

$A = .2$
$R = 3$: without Gateway retransmission
∞ : with Gateway retransmission
$k_1,k_2 = 25$
$R_1,R_2 = 1.5$
$1/u_1,1/u_2 = .5$

The message inter-arrival time A corresponds to the production (and transmission) of 5 messages in an average end-to-end subnet delay. This represents bulk traffic on a connection where the packet transmission delay, into (and out of) the Catenet, occupies a small fraction of overall end-to-end delay - e.g. where several packet switches or a satellite link separates the intermediate Gateway from either Host. The retransmission timeouts - R,R_1,R_2 - are set to 3 times the average transit medium delay of the level over which they operate. This value has been found to minimise retransmission delay, subject to minimising the number of retransmission duplicates produced, in this type of simulated connection (Edge 78). A wide range of packet loss rates is used, representing connections varying from the very reliable to heavily congested ones with frequent packet discard from Gate Halves or in each subnet.

The Endpoint and Hop-by-Hop approaches are represented by the following protocol mixes (Section 5.1):

Hop-by-Hop = Nil(SPAR,SPAR)
Endpoint with Gateway retransmission = SPAR,(PAR,PAR)
Endpoint without Gateway retransmission = SPAR,(Nil,Nil)

The use of Gateway level retransmission in the Endpoint approach has been proposed to improve efficiency over lossy nets (Binder 75, Walden 75), although it does require additional Gateway storage to keep packet copies for retransmission. Analysis (Metcalfe 73) has shown, for example, that such intermediate retransmission reduces end-to-end delay when packet loss is significant. A few of the results presented here demonstrate this and other drawbacks of not using Gateway level retransmission. Comparison with the Hop-by-Hop approach subsequently assumes the use of Gateway retransmission with the Endpoint approach, whenever packet loss is significant.

Figure 5 shows the variation of the average end-to-end message delay (from message appearance at the Source Host to delivery to the Destination Process) against the packet loss (p_1 and p_2) in each subnet. We note that analytical validation for the Endpoint approach without Gateway Retransmission is exact; whereas approximations made in the analysis of the other approaches (see Appendix 2) lead to small differences between corresponding analytical and simulation results. Figure 5 demonstrates the higher delay in the Endpoint approach when Gateway retransmission is not used, that we mentioned above. This occurs because end-to-end retransmission has to use a larger retransmission timeout than Gateway level retransmission. Comparison of the Endpoint approach, using Gateway retransmission, with the Hop-by-Hop approach shows the latter produces slightly larger delay, when packet loss occurs. Although both of these retransmit at the Gateway level with identical timeouts, the Hop-by-Hop approach sequences packets at the intermediate Gateway (between either net) in addition to Destination sequencing. As mentioned in Section 4, this imposes additional unnecessary delay.

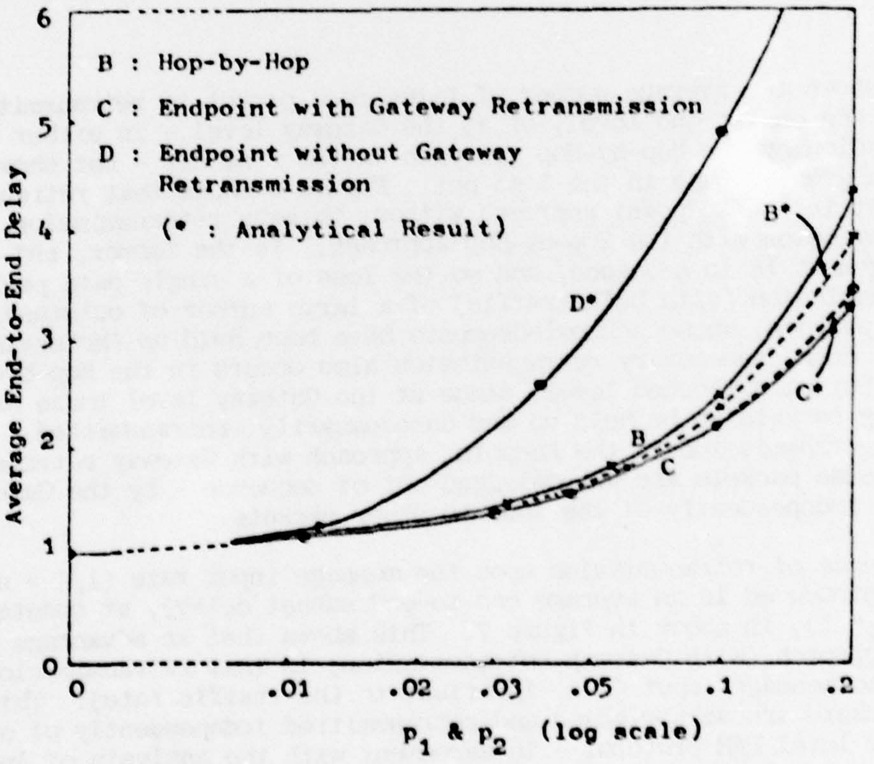


Figure 5. Average End-to-End Delay versus Packet Loss

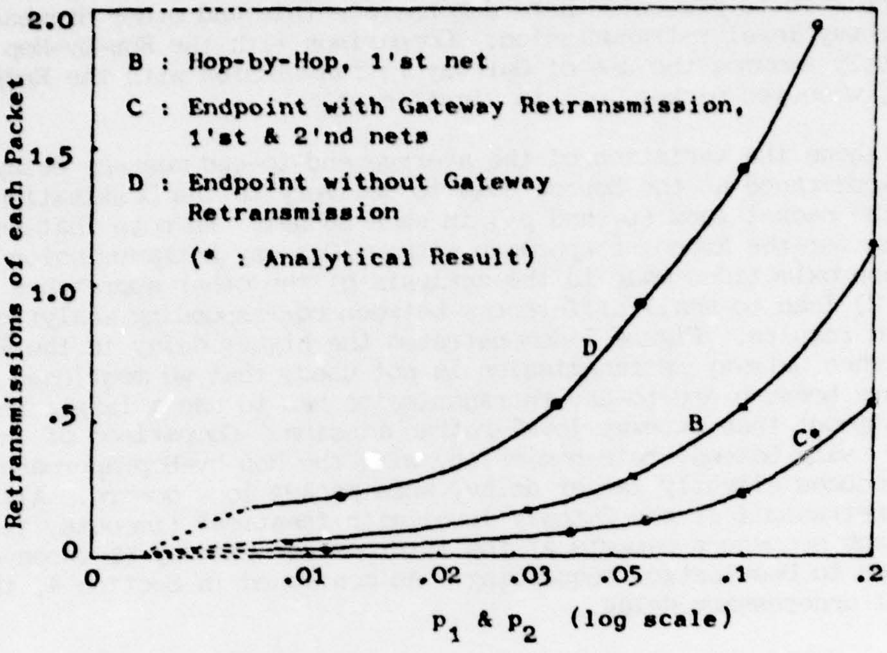


Figure 6. Average Retransmissions of each Packet versus Packet Loss

Figure 6 shows the average number of times each packet is retransmitted - either at the end-to-end level, or at the Gateway level - in either net. Retransmission by the Hop-by-Hop approach in the 2'nd net - not shown - is marginally greater than in the 1'st net. Figure 6 shows that retransmission is greatest in the Endpoint approach without Gateway retransmission, followed by retransmission with the Hop-by-Hop approach. In the former, end-to-end acknowledgement is in sequence, and so tne loss of a single data packet causes the retransmission (with bulk traffic) of a large number of outstanding successor packets, whose acknowledgements have been held up (McKenzie 74, Edge 78). Such unnecessary retransmission also occurs in the Hop-by-Hop approach, but at a reduced level, since at the Gateway level there are fewer outstanding packets to be held up and unnecessarily retransmitted. The low level of retransmission in the Endpoint approach with Gateway retransmission occurs because packets are acknowledged out of sequence - by the Gateway PAR protocol - independently of the loss to other packets.

The dependence of retransmission upon the message input rate (1/A = number of messages introduced in an average end-to-end subnet delay), at constant packet loss ($p_1=p_2=.1$), is shown in Figure 7. This shows that an advantage of the Endpoint Approach (with Gateway retransmission) is that retransmission is invariant to message input (i.e. invariant to the traffic rate). This occurs because packets are acknowledged and retransmitted independently of others by the Gateway level PAR protocol - in agreement with the analysis of Appendix 1. Conversely, retransmission in the Hop-by-Hop approach increases as the message input rate rises (i.e. as traffic becomes heavier) because on each packet loss, there are more outstanding packets available to be needlessly retransmitted (by the mechanism described earlier). It is only at very low message input ($1/A < 1$) - characteristic of interactive traffic - that lost packets can be

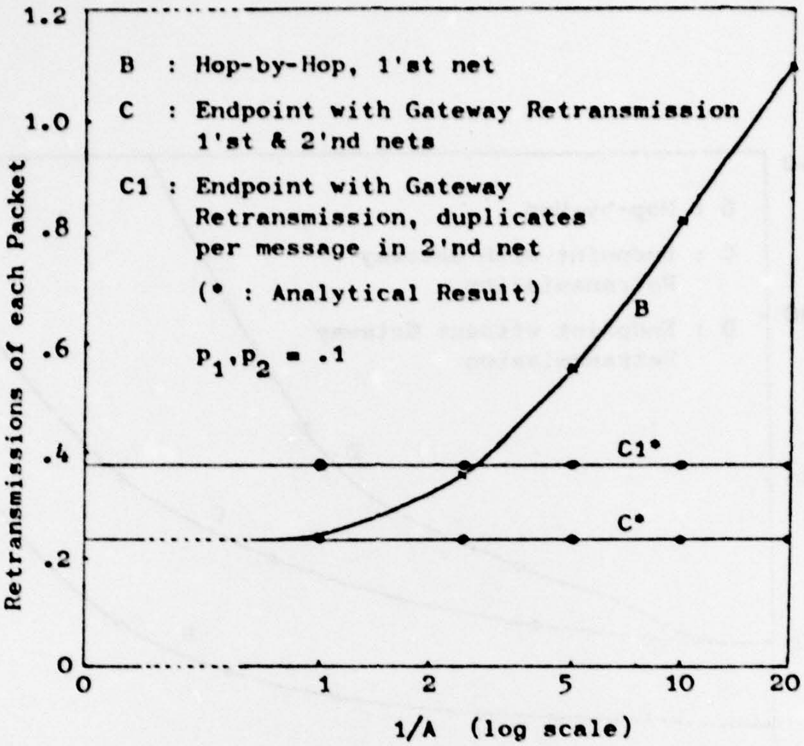


Figure 7. Average Retransmissions of each Packet versus Message Input Rate

retransmitted and acknowledged before later packets timeout. In this case, the retransmission rate approaches that of the Endpoint approach.

We note that in the Endpoint approach, Gateway retransmission duplicates from the first net can enter the second net, thereby increasing the average number of message duplicates therein. The average number of message duplicates per message in the second net is shown in Figure 7. Use of a simple duplicate filtering scheme at Gateways, as used between nodes in the ARPA Packet Radio Net (Kunzelman 78), would correct such "amplification" of duplicates.

In Figure 8, the total average storage requirement (for data packets) for both Hosts is shown, in terms of numbers of packets. This comprises retransmission queue storage at the Source Host and packet re-ordering storage at the Destination Host, for both the Gateway and end-to-end protocol levels. However, the Source Host is assumed to keep only one packet copy for retransmission at either level, when both of these levels operate. Figure 8 shows that Host storage requirement with the Hop-by-Hop approach is almost half that in the Endpoint approach, for low packet loss. This is to be expected since Host storage in the Hop-by-Hop approach is required only for a single net connection, and one would expect an even larger difference for connections spanning more than two nets.

We note that the occurence of excessive Host storage requirement, with high packet loss, for the Endpoint approach without Gateway retransmission, in Figure 8, is due to the long delay of end-to-end retransmission (noted earlier), which allows large queues of unacknowledged packets to accumulate at each Host, following packet loss.

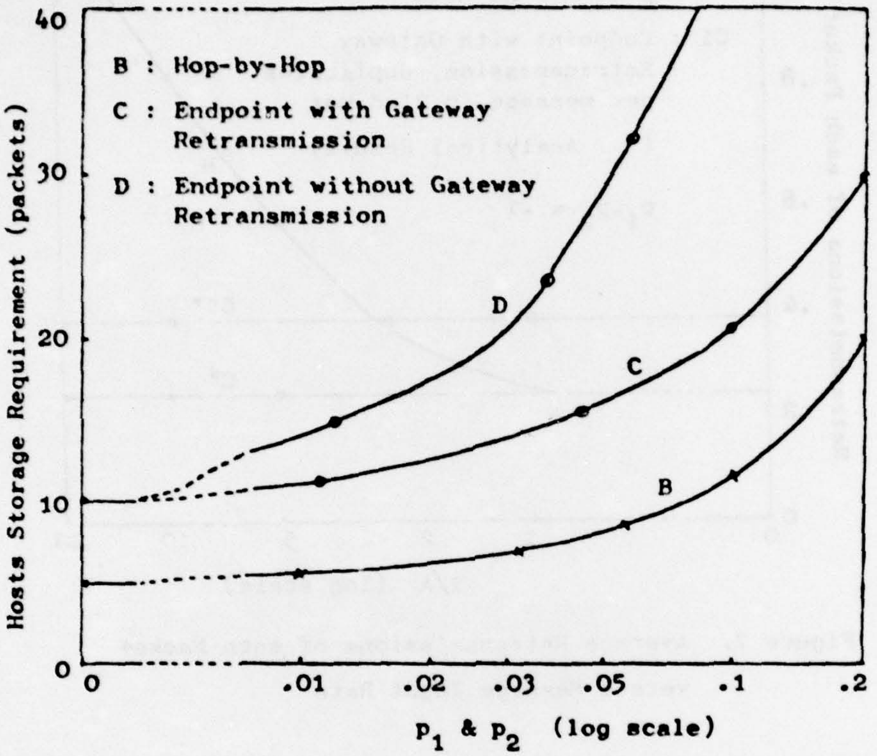


Figure 8. Average Hosts Storage Requirement versus Packet Loss

The average storage requirement (for data packets) in the intermediate Gateway, which separates the 1'st and 2'nd nets, is shown in Figure 9. This comprises packet re-ordering storage for the Gateway level across the 1'st net (Hop-by-Hop approach only) and retransmission storage for the 2'nd net (both approaches). Storage requirement is shown to be slightly greater in the Hop-by-Hop approach, when packet loss is significant. It would also be greater at low packet loss, since Gateway retransmission - and thus Gateway storage - would then not be required for the Endpoint approach. Such higher storage requirement illustrates the greater involvement of Gateways in the Hop-by-Hop approach.

Figure 10 combines the two previous results, and shows the average storage requirements both for the Hosts and for the intermediate Gateway. The value of this result depends upon the "costs" of Gateway and Host storage being equal (if they are not, each storage component must be weighted accordingly). When costs are equal, Figure 10 shows that the Hop-by-Hop approach is less costly - storage-wise - than the Endpoint approach, for non-zero packet loss.

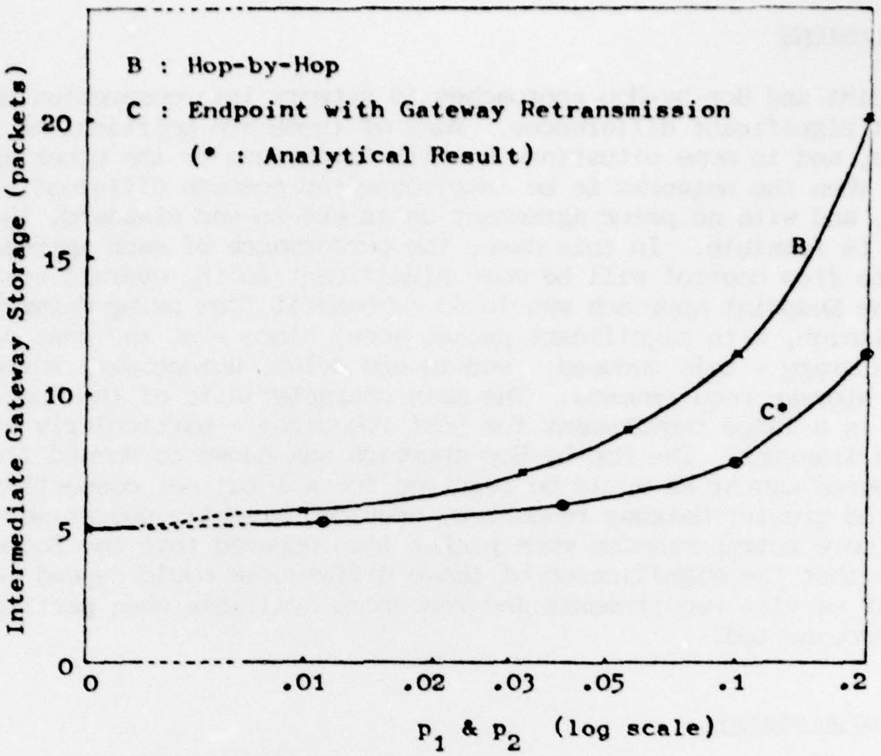


Figure 9. Average Intermediate Gateway Storage Requirement versus Packet Loss

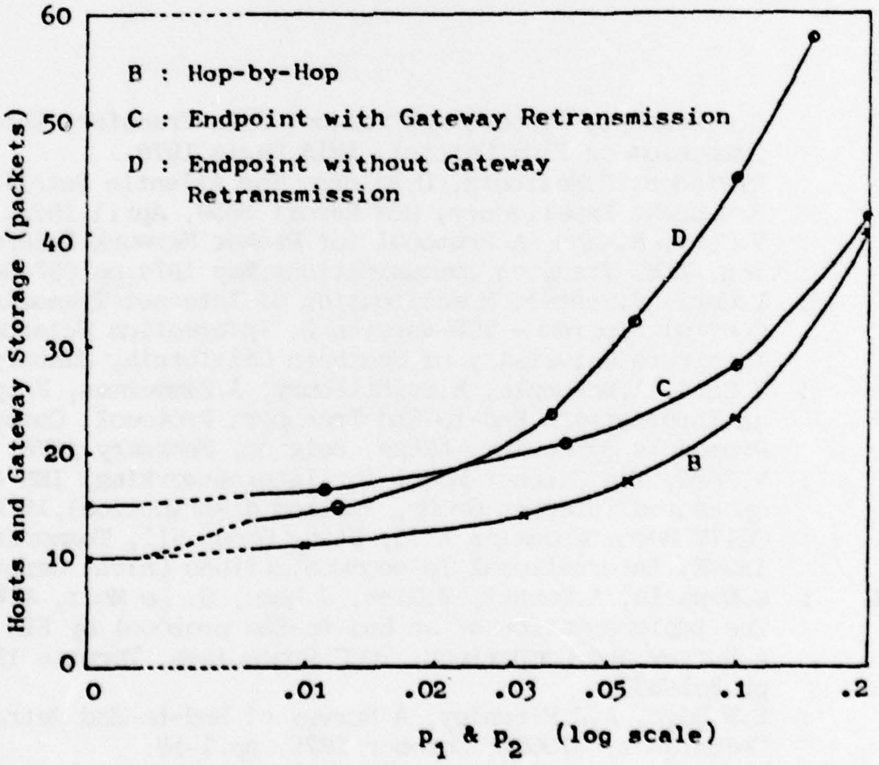


Figure 10. Average Hosts and Gateway Storage Requirement versus Packet Loss

7. CONCLUSIONS

The Endpoint and Hop-by-Hop approaches to network interconnection exhibit a number of significant differences. Many of these are political or technical in nature, and in some situations heavily favour one or the other approach. However, when the networks to be interconnected possess different virtual call services, and with no prior agreement on an end-to-end standard, then either approach is feasible. In this case, the performance of each approach with respect to flow control will be more significant to the overall evaluation of each. The Endpoint approach was found to benefit from using Gateway level retransmission, with significant packet loss, since - at the cost of extra Gateway storage - this reduced: end-to-end delay, unnecessary retransmission, and Host storage requirements. The main characteristic of the Endpoint approach is a large requirement for Host resources - particularly storage for bulk data transfer. The Hop-by-Hop approach was shown to demand from hosts the same resource amount as would be required for a local net connection. However, it required greater Gateway resources, provided slightly poorer service, and produced more retransmission when packet loss occured than the Endpoint approach. We expect that the significance of these differences would depend on the particular service requirements and resources available when particular nets were interconnected.

8. ACKNOWLEDGEMENTS

The author wishes to acknowledge the support of the Science Research Council; and is grateful to Mr.A.J.Hinchley, Professor P.T.Kirstein and other members of the INDRA Group for helpful comments, during the preparation of this paper.

REFERENCES

Bennett 79 : C.J.Bennett, Supporting Transnet File Transfer, International Symposium on Flow Control, IRIA,Paris,1979.

Binder 75 : R.Binder, R.Rettberg, D.Walden, The Atlantic Satellite Broadcast Experiments, BBN Report 3056, April 1975.

Cerf 74 : V.Cerf, R.Kahn, A Protocol for Packet Network Intercommunication, IEEE Trans on Communications,May 1974,pp 637-648.

Cerf 78a : V.Cerf, J.Postel, Specification of Internet Transmission Control Program - TCP version 3, Information Sciences Institute University of Southern California, January 1978.

Cerf 78b : V.Cerf, A.McKenzie, R.Scantlebury, A.Zimmerman, Proposal for an Internetwork End-to-End Transport Protocol, Computer Network Protocols Symposium, Liege, Belgium, February 1978, p.H-5.

Cerf 78c : V.Cerf, The Catenet Model for Internetworking, IEN 48 (DARPA-sponsored Internet Group, limited distribution),1978.

CCITT 78 : CCITT Recommendation X.75, Study Group VII, Temporary Document 132-E, International Telecommunications Union, Geneva,1978.

Deparis 76 : M.Deparis, A.Duenki, M.Gien, J.Laws, G. Le Moli, K.Weaving - The Implementation of an End-to-End protocol by EIN Centres: A Survey and Comparison, ICCC Procedings, Toronto 1976, pp.351-360.

Edge 78 : S.W.Edge, A.J.Hinchley, A Survey of End-to-End Retransmission Techniques, SIGCOM, October 1978, pp.1-18.

Gien 75 : M.Gien, J.Laws, R.Scantlebury, Interconnection of packet switching networks: theory and practice, Communication Networks, Online Conferences Ltd., Uxbridge, England,1975, pp.241-260.

Gien 78 : M.Gien, J.L.Grange, Performance Evaluations in CYCLADES, ICCC Proceedings, Tokyo, 1978, pp.23-32.

Higginson 75 : P.L.Higginson, A.J.Hinchley, The problems of linking several networks with a gateway computer, Communication Networks, Online Conferences Ltd, Uxbridge, England, 1975, pp.25-34.

Higginson 78 : P.L.Higginson, Z.Z.Fisher, Experiences with the initial EPSS Service, Eurocomp 78, Online Conferences Ltd, Uxbridge, England, 1978, pp.581-600.

Jacobs 78 : I.M.Jacobs,R.Binder,E.V.Hoversten, General Purpose Packet Satellite Networks, IEEE Trans on Communications,November 1978.

Kirstein 78 : P.T.Kirstein,C.J.Bennett - SATNET and the Provision of Transnet Service, Indra Note 674, Dept. of Statistics and Computer Science, University College London.

Kleinrock 77 : L.Kleinrock,H.Opderbeck - Throughput in the ARPANET -Protocols & Measurement, IEEE Trans on Communications,January 1977, pp.95-104.

Kunzelman 78 : R.C.Kunzelman, Overview of the Arpa Packet Radio Experimental Network - IEEE CompCon, San Francisco,Spring 1978, p.157.

Little 61 : J.D.C.Little - A Proof of the Queueing Formula L=λW,Operations Research 9,1961, pp.383-387.

Lloyd 75 : D.Lloyd,P.T.Kirstein, Alternative Approaches to the Interconnection of Computer Networks - Communication Networks,Online Conferences Ltd, Uxbridge,England, September 1975, p.499.

McKenzie 74 : A.McKenzie, Internetwork Host-to-Host Protocol - INWG General Note 74,December 1974.

McQuillan 77 : J.M.McQuillan,D.C.Walden, The ARPA Network Design Decisions, Computer Networks, Vol.1, No.5, August 1977, p.243.

Metcalfe 73 : R.M.Metcalfe, Packet Communication, MAC TR-114, MIT,Project MAC (Ph.D, Thesis), December 1973.

Metcalfe 76 : R.M.Metcalfe,D.R.Boggs, Ethernet: Distributed Packet Switching for Local Computer Networks - Communications of the ACM, Vol.19 No.7, July 1976,pp.395-404.

Postel 78 : J.B.Postel, Internetwork Protocol Specification, Information Sciences Institute, University of Southern California,June 1978.

Pouzin 74 : L.Pouzin, A Proposal for Interconnecting Packet Switching Networks, Eurocomp proceedings,May 1974,pp.1023-1036.

Pouzin 76 : L.Pouzin, Flow Control in Data Networks - Methods and Tools, ICCC Proceedings, Toronto 1976,pp.467-474.

Rybczynski 78 : A.M.Rybczynski,D.F.Weir,I.M.Cunningham, Datapac Internetworking for International Services, ICCC Proceedings,Tokyo 1978, p.47.

Roberts 73 : L.G.Roberts,B.C.Wessler - The ARPA Network, Computer Communication Networks, editors N.Abramson,F.Kuo,Prentice-Hall, 1973.

Sunshine 75 : C.A.Sunshine, Interprocess Communication Protocols for Computer Networks, TR-105, Digital Systems Lab, Stanford University, (Ph.D. Thesis), December 1975.

Sunshine 77a : C.A.Sunshine, Interconnection of Computer Networks, Computer Networks Vol.1, No.3, January 1977, pp.175-195.

Sunshine 77b : C.A.Sunshine, Efficiency of Interprocess Communication Protocols for Computer Networks, IEEE Trans on Communications,February 1977, pp.287-293.

Walden 75 : D.C.Walden,R.D.Rettberg, Gateway Design for Computer Network Interconnection -Communication Networks,Online Conferences Ltd., Uxbridge,England, September 1975, pp.113-128.

APPENDIX 1 - ANALYSIS OF SINGLE NET PERFORMANCE

Analysis of the performance of a single net, under both the PAR and SPAR protocols of Section 5.1, has already been carried out (Sunshine 75, Sunshine 77b). Here, we briefly outline the derivation of results relevant to the simulation model of Section 5.

We look at a single protocol level - PAR or SPAR - operating between a Sender and Receiver of packets, at each end of a single net - Figure 11. An arbitrary probability density function (pdf) f for the subnet delay is assumed, and its corresponding probability distribution function (PDF) will be denoted by F. A suitable unit of time for the system would be the mean of this delay. However, such normalisation requires modification in the multi-net case (see Appendix 2). We successively add in the effects to packet delay (measured from initial transmission at the Sender to delivery, by the Receiver, to the next stage) of: a packet loss probability q, repeated retransmission (of unacknowledged packets) at intervals I, and packet sequencing at the Receiver (SPAR only). We assume a constant interval T between successive packet arrivals (and hence between successive packet transmissions) at the Sender.

The subnet delay pdf can be modified to take account of packet loss by introducing an impulse function for infinite packet delay - i.e. for packet loss - (Sunshine 75). We denote the resulting pdf fq and PDF FQ for packet delay (x) as follows:

$$fq(x) = \lim_{c \to \infty} [(1-q).f(x).(1-H(x-c)) + (1-q).(1-F(c)).d(x-c) + q.d(x-c)] \quad x>=0 \qquad (2)$$

$$FQ(x) = \lim_{c \to \infty} [(1-q).F(x).(1-H(x-c)) + H(x-c)] \quad x>=0 \qquad (3)$$

with d = Dirac Delta Function
H = Heavyside Unit Step Function

$$H(y) = \begin{cases} 1 : y>=0 \\ 0 : y<0 \end{cases}$$

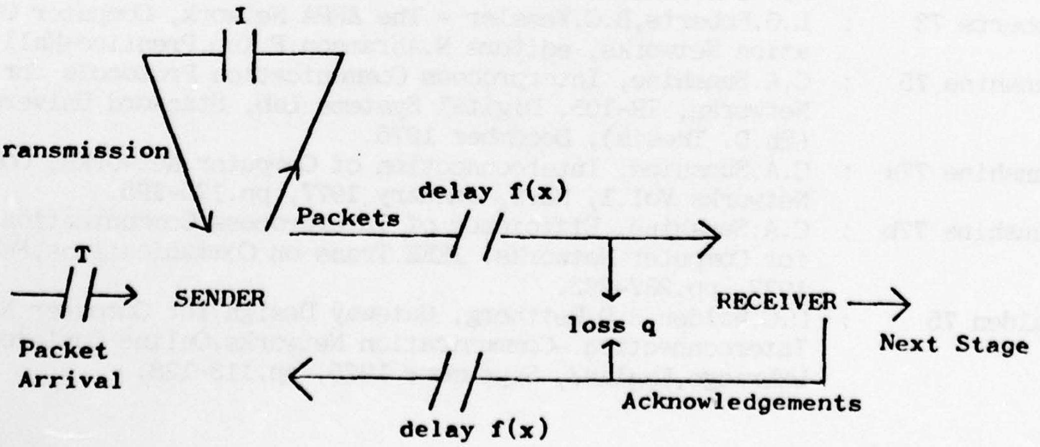


Figure 11. Single Net under PAR or SPAR protocol

The pdf g and PDF G for delay with retransmission are:

$$G(x) = 1 - \prod_{k=0}^{n}(1 - FQ(x-k.I)) \qquad (4)$$

$$g(x) = \sum_{j=0}^{n} fq(x-j.I).\prod_{\substack{k=0\\k\neq j}}^{n}(1 - FQ(x-k.I)) \qquad (5)$$

$$\text{with } n = \lfloor x/I \rfloor$$

The PDF H for delay, with sequencing at the Receiver, is:

$$H(x) = \prod_{j=0}^{\infty} G(x + j.T) \qquad (6)$$

The above analysis also applies to packet round trip delay - for travel to the Receiver and back to the Sender (e.g. as an acknowledgement on the return trip) - provided q, f, F, fq, FQ, g, G in eq's 2 to 5 are replaced by their round trip counterparts, which we denote by adding a dash to each.

The following performance measures can be derived from the above results:

av. PAR 1-way delay $= \int_0^{\infty}(1 - G(x)).dx$ (7)

av. PAR round trip delay $= \int_0^{\infty}(1 - G'(x)).dx$ (8)

av. number of retransmissions per packet with PAR $= \sum_{j=1}^{\infty}(1 - G'(j.I)$ (9)

av. number of packets in PAR retransmission queue = av. PAR round trip delay/T (10)

av. SPAR 1-way delay $= \int_0^{\infty}(1 - H(x)).dx$ (11)

av. number of packets at SPAR Receiver = (av. 1-way SPAR delay - av. 1-way PAR delay)/T (12)

Equation 8 gives the average packet acknowledgement delay for the PAR protocol. Equations 10 and 12 use Little's result (Little 61) for the average number of customers in a queueing system, given their average waiting time and arrival rate. We note that none of the performance measures obtained for the PAR protocol requires constant inter-packet arrival time at the Sender (T in eq'10 can be the average inter-arrival time); and the PAR retransmission overhead (eq'9) is independent of T (i.e. independent of the traffic rate).

APPENDIX 2 - ANALYSIS OF MULTI-NET PERFORMANCE

Here, we look at the simulation model of Section 5, which comprises "n" concatenated nets, with a single end-to-end protocol level and a Gateway protocol level across each net. When the Gateway level is not used, the end-to-end level effectively operates over a single net (whose delay is the sum of the delays over the n individual nets), and performance measures may be obtained from Appendix 1. Otherwise, there are n concatenated PAR or SPAR stages, with an end-to-end protocol over them - Figure 12. The average inter-packet arrival interval at the i'th stage is denoted by A_i. Here, a suitable unit of time for the system is the sum, over the n stages, of the mean subnet delay of each stage (in the absence of packet loss). Such normalisation is used, for example, in Section 6.

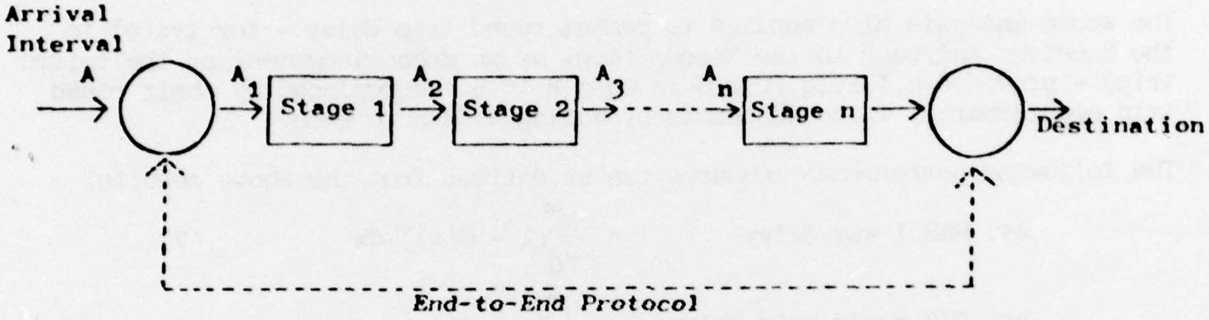


Figure 12. End-to-End Protocol over n PAR/SPAR Stages

We look first at the protocol mix SPAR(PAR,PAR, ... , PAR) - used to represent the Endpoint approach with Gateway retransmission in Section 6. We assume there is no end-to-end level retransmission - this being done at the Gateway level. The analysis of Appendix 1 applies to each PAR stage, with the parameters of the simulation model (Section 5.2) substituting for corresponding parameters in Appendix 1. The average inter-packet arrival interval A_{i+1} at stage i+1 is given by:

$$A_{i+1} = A_i/[(1 - p_i) \times (1 + \text{av. retransmissions per packet in stage i (eq'9)})]$$

with $A_1 = A$ (= av. inter-message arrival interval, Section 5.2)

p_i = Packet loss probability in the i'th net (Section 5.2)

The above expression takes into account the production of retransmission duplicates by each PAR stage, and allows the determination of the average retransmission queue size (eq'10) and the average number of message duplicates for each PAR stage (which for the i'th stage is: A/A_i x retransmission overhead for the i'th stage, from eq'9).

The overall end-to-end delay, and the average number of packets being re-sequenced at the Destination Host (by the end-to-end SPAR protocol), may be obtained reasonable simply, if the receiving Gate-Half of each PAR stage is assumed to filter out (retransmission) packet duplicates (giving: A_i=A, for all i). The end-to-end delay derived thus, will be slightly larger than without such duplicate filtering, since message duplicates increase the chance of early arrival at the Destination. However, for low packet loss (and hence low duplicate production), these results serve for model validation. The probability density function (pdf) gn, and probability distribution function (PDF) GN for the end-to-end delay through stages 1 to n, are given by the convolution of the pdf's for (one-delay) delay across each PAR stage:

$$gn(x) = \int_0^x g_1(x-x_1)\int_0^{x_1} g_2(x_1-x_2) \;\ldots\; \int_0^{x_{n-2}} g_{n-1}(x_{n-2}-x_{n-1}).$$

$$g_n(x_{n-1}).dx_{n-1}dx_{n-2} \;\ldots\; dx_1$$

$$GN(x) = \int_0^x gn(u).du$$

with g_i = g from eq'5 (Appendix 1) for the i'th stage

The following performance measures can be obtained from GN:

PDF of end-to-end delay with sequencing $= HN(x) = \prod_{j=0}^{\infty} GN(x+j.A)$ (13)

av. unsequenced end-to-end delay $= \int_0^{\infty}(1 - GN(x)).dx$ (14)

av. sequenced end-to-end delay $= \int_0^{\infty}(1 - HN(x)).dx$ (15)

av. no. packets being sequenced at Destination = (av. sequenced end-to-end delay - av. unsequenced end-to-end delay)/A (16)

with A = inter-message arrival interval (Section 5.2)

We note that equation 13 expresses the requirement that for a packet to be in sequence at the Destination, it and all its predecessors must have arrived. Equation 16 uses Little's result (Little 61) for average customer queue size, given the average queueing delay and average customer arrival rate.

We now look at the protocol mix Nil(SPAR,SPAR, ... , SPAR) - used for the Hop-by-Hop approach in Section 6. Here the average inter-packet arrival interval at each stage is A, since each SPAR stage filters out duplicates. The performance for the first stage can be obtained from Appendix 1, since for the first stage (only), packets arrive at constant intervals. Performance for the other stages can be obtained approximately (from Appendix 1), by assuming that packet arrivals at these are also at constant intervals. The average end-to-end message delay is given approximately by the sum of the delays for each stage, obtained in the above way. Such approximate results serve to partially validate corresponding simulation results.

7.3 A Survey of End-to-End Retransmission Techniques

A SURVEY OF END-TO-END RETRANSMISSION TECHNIQUES

S.W. EDGE & A.J. HINCHLEY

Department of Statistics and Computer Science
University College London

1. Introduction

Retransmission of lost or damaged messages from a sender to a receiver is a basic ingredient of computer network protocols. It can occur at many levels from a simple point-to-point line level to an end-to-end connection across a number of levels of data path, where several mechanisms combine to ensure reliable data transfer.

Specific instances of retransmission are found at line level in protocols such as HDLC (CCITT 76), and between packet switches in a network (McQuillan 77). In a multiple network environment, retransmission might be employed between gateways operating at the edges of the individual networks (Sunshine 77a). Finally, a reliable delivery end-to-end protocol may support process-process communication across a one - or many - network path, and such a protocol may also require a retransmission capability. Examples of the latter are: TCP (Cerf 78a), INWG 96 (Cerf 78b), the EIN end-to-end protocol (EIN 76), and the CYCLADES end-to-end protocol (CYCLADES 73).

Other instances of retransmission occur in packet switching over a broadcast medium, and specialised retransmission schemes have evolved for use in broadcast satellite operation (Binder 75), packet radio (Kunzelman 78), and with Ethernet (Metcalfe 76).

This paper concentrates on the aims of end-to-end retransmission. In particular we look at two alternative schemes: positive acknowledgement of data coupled to a sender timeout, which is well-researched, and use of negative acknowledgement, which is not. These are reviewed in terms of a defined set of retransmission objectives, and the advantages and disadvantages of each scheme are demonstrated using a simulation model of a simple end-to-end connection.

We note that in our terminology, a "transport station" will denote the physical implementation of an end-to-end protocol at a particular site, and end-to-end communication will be between pairs of "processes".

2. Aims of End-to-End Retransmission

The aims of a retransmission mechanism are:

ensuring reliability
minimising delay
minimising redundant duplicate retransmissions
simple operation

These aims relate to any level of protocol, but particularly at the end-to-end level, where both the delay and variation in delay are scaled up to such a degree that meeting the criteria mentioned above becomes rather more of a critical matter.

The minimisation of delay is important in packet switched networks, where nodal switching delays add up to a significant amount compared to say the delay in a pre-established digital circuit. For example, the transit delay of existing X25 networks is typically one sixth to half a second (Erskine 77, Guilbert 77). Howevever, retransmissions, which will normally be keyed to some message (or lack of it) from the receiver, are likely to be delayed by at least two such intervals.

Minimisation of retransmission overhead is an obvious criteria related to efficient use of resources - both communication resources to carry the unnecessary additional messages, and processor resources in generating and interpreting the messages. In public networks, packet costs will be incurred for such messages, and in large private datagram networks, runaway retransmissions may cause network congestion, which is not easily recoverable. We note that efficient resource utilisation and minimisation of delay are also the goals of flow control (Pouzin 76).

Finally, simplicity is important, both to allow unambiguous definition and to reduce the size and complexity of transport stations.

The diversity of these aims argues against a single scheme for all situations. By broadly classifying different end-to-end reliability levels it can be shown, however, that schemes can be introduced to satisfy the different requirements.

3. Positive Acknowledgement Retransmission

Positive acknowledgement retransmission, using a timeout at the sender, is the most common retransmission scheme adopted. Its main advantage is simplicity: data made available to a receiving process is positively acknowledged by the receiving station; data which times out at a sending station is retransmitted. Reliability is virtually guaranteed in all circumstances short of system crashes, which might permanently remove the necessary state information to maintain the connection (Sunshine 75). In addition, however, careful re-use of either message sequence numbers (Tomlinson 74, Dalal 74) or connection identifiers (Reed 77) will be necessary, in order to reliably detect message arrivals from previous (recent) incarnations of a connection.

Analysis (Sunshine 75) has shown the delay incurred by the need to rely, on occasions, on retransmitted packets, is reduced by using a smaller retransmission timeout. Minimum retransmissions, however, are achieved by employing a timeout equal to twice the maximum packet life time for the communication path (if known), thus ensuring that retransmissions are only triggered when a packet is definitely lost or damaged. Since the retransmission delay with such a timeout could be excessive, a smaller "tuned" timeout might be used, where the probability of retransmitting a packet, about to be acknowledged, is kept very small. The value of such a tuned timeout will always exceed the average round trip delay of the transit medium - to avoid runaway retransmission - but the excess could be small (Sunshine 75).

The most serious drawback of positive acknowledgement retransmission is that, under certain conditions, a very high level of redundant duplicate retransmission is unavoidable (McKenzie 74). This is possible whenever two or more data carrying packets are pipelined (i.e. simultaneously outstanding) at a sender. Because these must be acknowledged in sequence (following data delivery to a receive process) the loss of a single packet delays the acknowledgement of all subsequent packets and unnecessarily induces their retransmission. The pipelining of a large number of packets may thus lead to a large ratio of retransmitted packets to packets actually lost. Further, in schemes where a large unit of acknowledgement is used - such as a letter in INWG 96 - unnecessary retransmission will be even higher, since in addition to succeeding packets, those packets preceeding a damaged packet, but belonging to the same acknowledgement unit, will also require retransmission.

The seriousness of the above obviously relates to the level of end-to-end packet loss. For very low loss rates, even when acknowledgement is per letter, the absolute magnitude of retransmission is likely to be low (Day 75). However, for high loss rates, alternative retransmission schemes may be needed, which reduce the level of redundant retransmission.

3.1 Modifications of Positive Acknowledgement Retransmission

We explore several modifications to positive acknowledgement retransmissions, which reduce the number of redundant retransmissions.

Several authors (McKenzie 74, Sunshine 75) have suggested retransmitting only the first packet timed out (repeatedly if necessary) on the assumption that for a low end-to-end loss rate, this will probably be the only packet lost. The main disadvantage of this scheme is that it would have a very high recovery delay whenever a large number of packets were discarded at a receiving station - the occurrence of which can be an important reason for requiring an end-to-end retransmission capability (Cerf 74). We therefore conclude that this is not a good general scheme.

In another scheme, which has been proposed for TCP (Mathis 77), the retransmission interval for each packet commences at some base value and increases linearly or exponentially following each retransmission of the packet. The main purpose is to avoid flooding the subnet and receiver with retransmissions, in the event that packets have to be discarded at the receiver (or in the subnet) and flow control alone is inadequate to quench this flow. We expect this scheme will be most useful with transport stations operating over datagram networks which implement minimal congestion control. It is interesting to note that an increasing retransmission interval may also be used in broadcast networks for a similar purpose (Metcalfe 73).

3.2 Selecting a Retransmission Timeout

As we noted above, the retransmission timeout used with positive acknowledgement is subject to two constraints: it should be large enough to minimise, or at least considerably reduce, the probability of premature retransmission and it should avoid unnecessarily high retransmission delays. The importance of the latter is obviously related to the frequency with which retransmission is required. Thus for extremely reliable end-to-end paths, such as occur across the Arpanet or across X25 nets, a single large timeout might be used at a transport station, which can be pre-set to minimise retransmission on all connections. For less reliable paths, such a large timeout would be unsuitable for those connections with a low transit delay. Instead, connections could be partitioned into categories - e.g. satellite, many hop terrestrial, etc. - based upon their order of round trip time, with a suitable timeout for each category. Finally, for very unreliable paths, the retransmission timeout might have to be tuned individually to each connection, in order to minimise retransmission delay. The following algorithm, which. continuously re-evaluates the retransmission timeout from round trip delay measurements, is one method of achieving this.

Each time that a new packet is acknowledged at a sending transport station, the round trip delay t since it was first transmitted is determined, using timestamping information associated with the packet retransmission queue. Then, provided the packet was not retransmitted - when t could be misleading - the average round trip delay estimate T, for the connection, is updated as follows:

$$T \ := \ (m.T + t)/(m+1) \quad m>=o \qquad (1)$$

The value of T obtained thus is affected by all the delays inherent to the connection, and is consequently more useful than knowledge of the delay of the transit medium alone. Appendix 1 shows that the weight m has two features: reduced m reduces the delay in correcting T when the connection round trip delay changes; but increased m reduces statistical fluctuation of T in steady state.

The retransmission interval can be periodically updated from the latest value of T:

Retransmission Timeout := n.T

The multiplier n, which should exceed 1, is set to ensure a low probability of premature retransmission. Its setting will thus depend on the likely fluctuation in acknowledgement delay, and could be small for low fluctuation.

4. Negative Acknowledgement Retransmission

Retransmission induced by explicit negative acknowledgement of lost packets by a receiver is far less common, at all levels, than positive acknowledgement retransmission. One example where it is used is in HDLC (CCITT 76), which uses two types of reject command to prompt retransmission (Gelenbe 78). In the end-to-end case, there is a proposal to incorporate negative acknowledgement in the INWG 96 protocol (Cerf 78b).

The major disadvantage of using negative acknowledgement in end-to-end protocols is the complexity this would add. For example, a receiving transport station would have to detect packets lost en route, possibly by timing-out missing message fragments. Furthermore, positive acknowledgement retransmission, with a suitably large timeout (Pouzin 73), would still be required to ensure reliability, in the event that negative acknowledgements or retransmissions were lost. Positive acknowledgement retransmission would probably also be required for interactive traffic, where loss of a small isolated message might go undetected at a receiving transport station. In the latter case, use of a small sender timeout to hasten positive acknowledgement retransmission could be integrated with negative acknowledgement by restarting the timeout of a packet, whenever it or a packet with lower sequence than itself was negatively acknowledged.

The main advantage of negative acknowledgement retransmission is to reduce redundant retransmission. For example, negative acknowledgement of missing "letter" fragments (see Section 5) might be used in INWG 96, in addition to positive acknowledgement of whole letters. With pipelined traffic, this additional means of re-supplying lost packets would substantially reduce the likelihood (discussed in Section 3) of unnecessarily retransmitting successors to - or packets in the same letter as - a damaged packet. As an alternative, the unit of positive acknowledgement in INWG 96 could be reduced - e.g. to letter fragments - to avoid retransmitting a whole letter whenever a letter portion was lost, but this would not avoid retransmitting packet successors when traffic was pipelined.

5. Comparison of Positive and Negative Acknowledgement Retransmission

We illustrate the points made above about each basic retransmission scheme with some simulation and analytical results for an example model of an end-to-end connection. The model, we will describe, comprises a single process-to-process connection maintained by a pair of transport stations (TS's), where data flow is one way from a Send Process (source) to a Receive Process (sink). The data transport mechanism, under review, is a simple one common to many end-to-end protocols, such as TCP(Cerf 77) and INWG 96 (Cerf 78b). Figure 1 illustrates the connection.

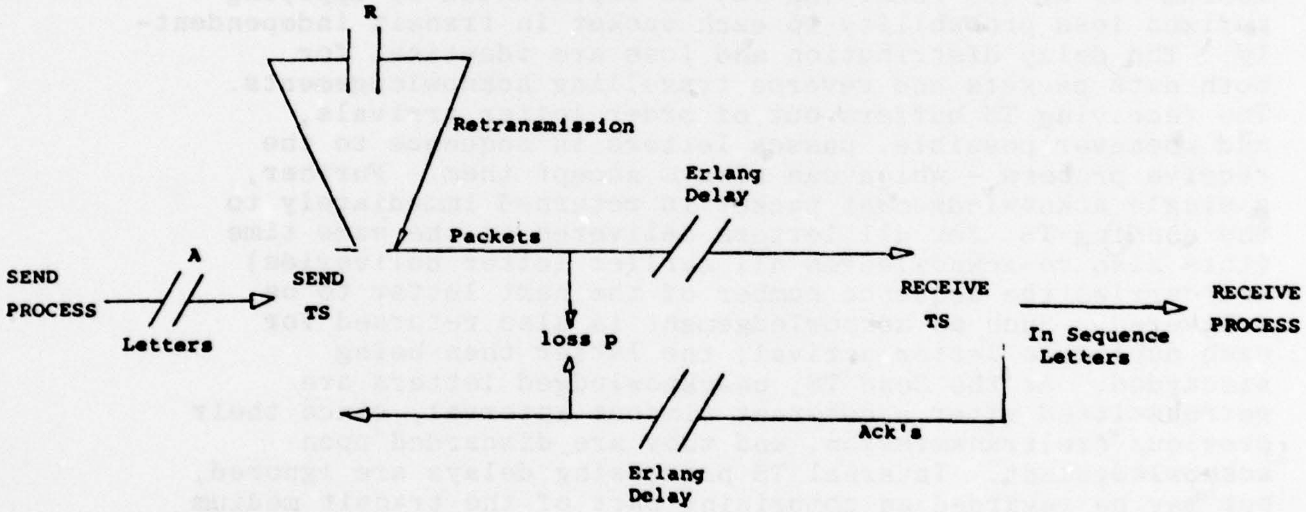


Figure 1. End-to-End Connection Model

An infinite stream of fixed size "letters" is passed one at a time and at constant time intervals from the Send Process to the Sending TS. Each letter is encapsulated in a single packet, which is assigned a monotonically increasing letter sequence number, and transmitted immediately. There is no restriction on flow, corresponding to a very large window and unlimited internal buffering in the case of TCP and INWG 96. Further, packets are not fragmented (e.g. at gateways) during transit. The transit delay between the two TS's has an Erlang distribution with parameter k and mean 1/u (which we set to unity). The probability density function f for a delay x is:

$$f(x) = (k.u)^{k}.x^{k-1}.e^{-kux}/(k-1)! \quad x >= 0 \qquad (2)$$

This models a wide range of delay distributions, from exponential (k=1) to constant (k approaches infinity). The coefficient of variation of delay is $k^{-\frac{1}{2}}$. We will use two values of k, k=25 and k=4, to model low and high delay variability respectively. Loss of packets in the transit medium (or at the receiving TS) is represented by applying a fixed loss probability to each packet in transit independently. The delay distribution and loss are identical for both data packets and reverse travelling acknowledgements. The receiving TS buffers out of order letter arrivals, and whenever possible, passes letters in sequence to the receive process - which can always accept them. Further, a single acknowledgement packet is returned immediately to the sending TS, for all letters delivered at the same time (this also re-acknowledges all earlier letter deliveries) and carries the sequence number of the next letter to be delivered. Such an acknowledgement is also returned for each duplicate letter arrival; the latter then being discarded. At the Send TS, unacknowledged letters are retransmitted after a constant timeout interval, since their previous (re)transmission, and they are discarded upon acknowledgement. Internal TS processing delays are ignored, but may be regarded as comprising part of the transit medium delay.

The unit of time is the mean TS-TS transit delay (1/u in eq.2). The model than has the following parameters:

k:	parameter of Erlang delay distribution
A:	inter-letter arrival interval at the send TS
p:	probability of packet loss in the transit medium
R:	send TS retransmission interval

The model described so far is used to illustrate positive acknowledgement retransmission. We illustrate negative acknowledgement with the following additional mechanism, which has been chosen for its simplicity. The Receive TS buffers and delivers arriving letters as before. However, each arriving letter commences a timeout period if its immediate sequential predecessor has not yet arrived. If the latter has not arrived on timeout, a negative acknowledgement is returned to the Send TS for all the immediately preceding non-arrivals - up to but not including the highest sequence preceding letter which has arrived. In practice, only the highest and lowest sequence numbers of these would be physically carried. Negative acknowledgements are also lost with probability p, representing say the case where acknowledgements are piggy-backed in a reverse data stream. At the Send TS, negatively acknowledged letters are immediately retransmitted, independently of positive acknowledgement retransmission (which operates as before). When implemented this scheme requires a new parameter:

Tnack = Timeout at the receive TS

The first example we illustrate with this model concerns the average number of times letters are retransmitted with positive acknowledgement retransmission operating only., for different values of the retransmission timeout R. Simulation results for this are shown in Fig.2, with A set to represent real-time on bulk transfer traffic. The first point to notice is that for a large enough R, retransmission is minimised for each case considered, as we stated earlier. Tuned R - discussed in Section 3 - is the minimum value of R giving this minimisation, and we will call this value Rtun. It clearly exceeds the average normalised round trip delay (which is 2) in every case, and is increased by higher delay variability (reduced k). R less than Rtun causes premature letter retransmission, and the rise of this (with reduced R) becomes more stepwise as acknowledgement delay approaches a constant (k approaches infinity, p approaches zero). Excessive retransmission for pipelined traffic - discussed in Section 3 - is illustrated for the case k=25, p=1, where the minimum level of retransmission at .54 substantially exceeds the minimum requirement of .11, when only lost packets are retransmitted.

Next we look at this situation from the point of view of negative acknowledgement. Figure 3 shows simulation results relating the value of the timeout Tnack used at the receiver to the extent of redundant retransmission when there is no packet loss (p=0). The results are analogous to those in Fig.2. A low value of Tnack increases the likelihood of prematurely negatively acknowledging letters, which have not arrived, thereby producing redundant retransmissions.

Conversely, for large enougn Tnack, redundant retransmission ceases. We can thus determine a tuned Tnack value to reduce the retransmission delay when packets are lost, in the same way that Rtun was found above. Figure 3 shows that its value depends on the variability of the transit medium delay (k) and the traffic rate (A), each of which affects the extent to which letters can arrive out of order.

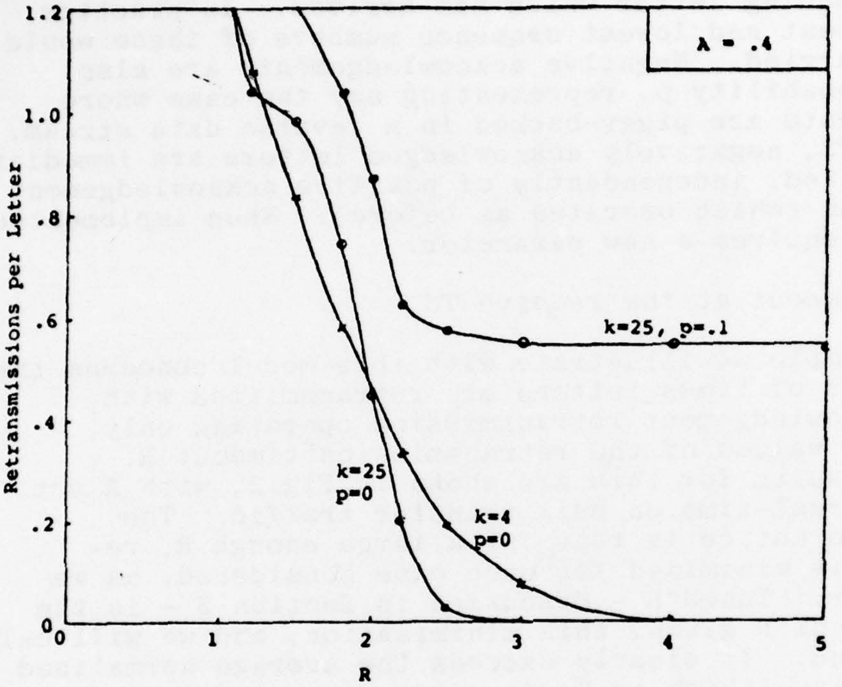


Figure 2. Average Retransmissions per Letter versus Sender Timeout

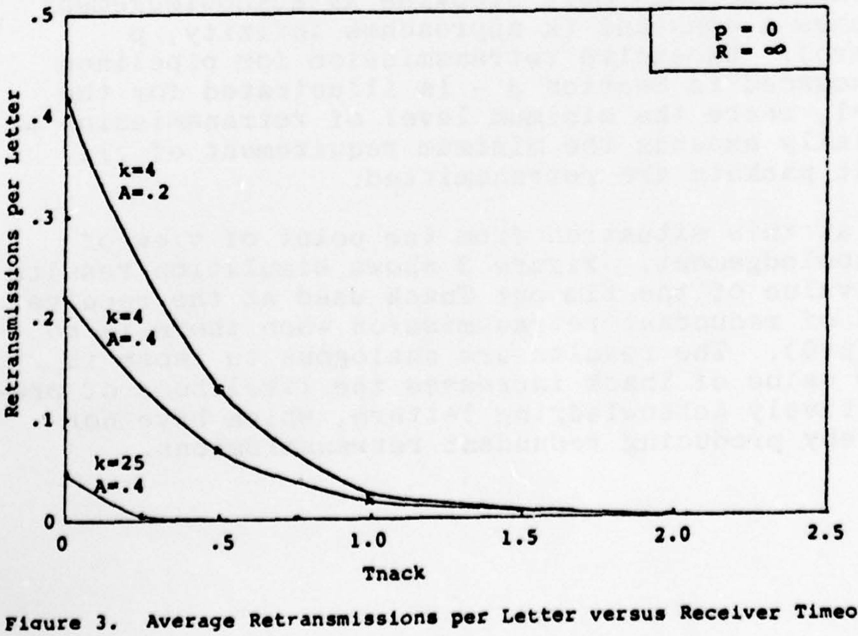


Figure 3. Average Retransmissions per Letter versus Receiver Timeout

The preceeding results allow comparison of the minimum (tuned) retransmission delay inherent in either scheme. For positive acknowledgement, this is simply Rtun. For negative acknowledgement, an approximate retransmission delay is obtained by assuming that the immediate sequential successor to a lost letter arrives safely at the receiver, and after timeout and return of a negative acknowledgement, induces the retransmission (i.e. p is small, k is large). Thus:

$$\text{av. retransmission delay} = \text{av. round trip delay} + \text{Tnack} + \text{T}$$

$$= 2 + \text{Tnack} + \text{T}$$

For the case p=O, k=25, A=.4, Rtun is 3 (Fig. 2), giving equality of retransmission delay for tuned Tnack =.6 (Fig. 3). The case p=O, k=4, A=.4 is similar, providing near equality.

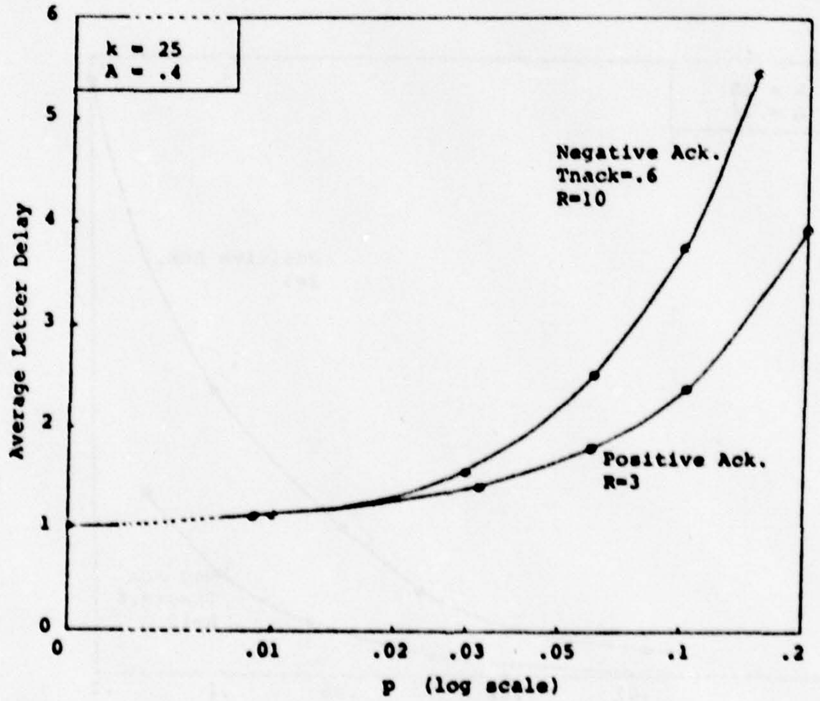


Figure 4. Average Letter Delay versus Packet Loss

The above comparison implies that delays in either scheme should be similar. We investigate this in Fig.4, which shows the average letter delay, from the Send to the Receive Process, against packet loss. Simulation results are shown for negative acknowledgement, whereas for positive acknowledgement we use analytical results, from Appendix 2, whose accuracy slightly exceeds that obtained from simulation. We re-use the tuned timeouts quoted above, so it is not surprising to find in Fig.4 almost identical delays for a low loss rate. At higher loss rates, the delay for negative acknowledgement becomes much larger, because substantially more lost packets require positive acknowledgement retransmission, following the loss of a negative acknowledgement or retransmission, and this uses a large untuned timeout (R=10). The latter setting reflects the reduced significance of the sender timeout with negative acknowledgement retransmission, although we would obviously expect to reduce delay by employing a smaller sender timeout.

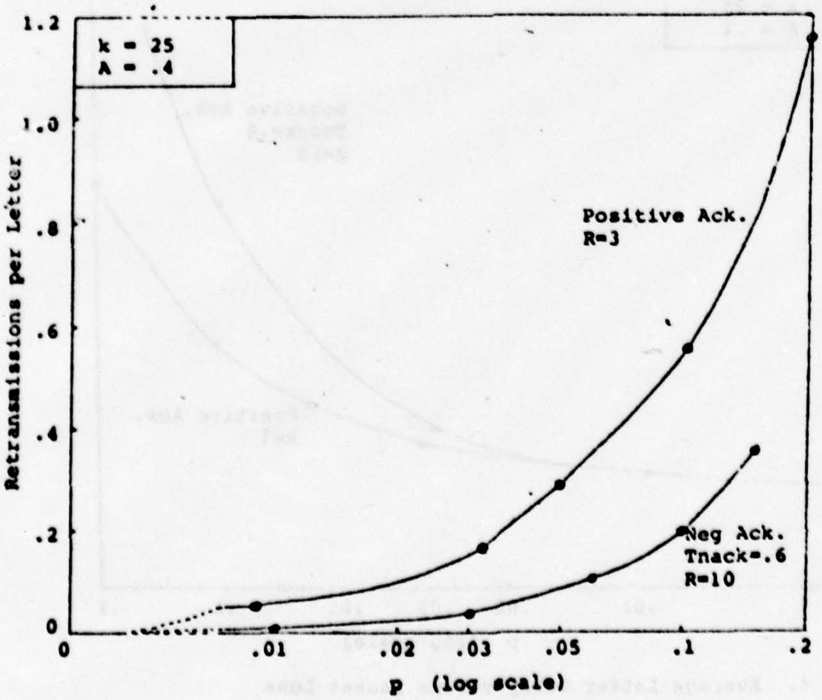


Figure 5. Average Retransmissions per Letter versus Packet Loss

In Fig. 5, we compare the retransmission overhead of each scheme - obtained by simulation - over the same range of loss rates used in Fig. 4. For low loss rates, retransmission in either scheme is low. However, it is clear that negative acknowledgement produces far fewer retransmissions than positive acknowledgement, and the magnitude of the difference becomes more significant for increased packet loss.

6. Selection of a Retransmission Scheme

We evaluate here the two basic schemes we have been comparing, in the context of three levels of end-to-end reliability.

6.1. Highly Reliable End-to-End Path

In this case we assume that lower level mechanisms provide reasonably reliable communication. An X25 virtual call network is an obvious example, where the bit error probability can be as low as 10^{-10} (Danet 76). Such a loss is acceptable for most network use, but where a guaranteed process-to-process reliability several orders of magnitude better is required, we may expect to superimpose an additional reliable end-to-end protocol. As an example, a version of the INWG 96 protocol adopted specifically for use above X25 networks is currently being produced by an IFIP working group (Cerf 78b).

It is clear that the frequency of necessary retransmission is sufficiently low that positive acknowledgement retransmission using a long timeout, will be adequate.

6.2. Moderately Reliable End-to-End Path

We reserve this case for end-to-end packet loss probabilities lying between 10^{-2} and 10^{-6}, the range likely to be found in most datagram networks, or where a receiving transport station may occasionally discard packets due to flow control (Cerf 74). Here retransmission will be required too infrequently to make the reduction in retransmission overhead, possible with negative acknowledgement, worthwhile. Positive acknowledgement will thus be suitable, and the value of the timeout may be set by categorising connections, as discussed in Section 3.2

6.3. Unreliable End-to-End Path

This case is for packet loss probabilities less than 10^{-2}. Such a high loss rate is obviously atypical, because in nearly all practical situations, lower level mechanisms reduce the loss rate seen at the end-to-end level. However, it may occur in special circumstances, such as where a receiving station uses inadequate buffering and must frequently discard arriving packets (Edge 77).

Provided a high level of retransmission overhead is acceptable, positive acknowledgement retransmission could be used here, and the retransmission timeout would probably require tuning - as described in Section 3.2 - to reduce delay. Alternatively, the simulation results discussed in Section 5 show that a negative acknowledgement scheme could substantially reduce the level of unnecessary retransmission at the cost of somewhat larger delay.

7. Conclusions

We conclude that positive acknowledgement retransmission - as employed in many existing end-to-end protocols - is most suitable for the majority of end-to-end connections, namely those with moderate or high reliability. For unreliable connections, negative acknowledgement retransmission may be preferable, because this greatly reduces the extent of redundant retransmission, which can be large with just positive acknowledgement. The drawbacks to negative acknowledgement are added complexity and probable higher delay. Optimal performance in either scheme requires careful selection of timeout values, and for unreliable connections, these should be tuned to the individual connection.

8. Acknowledgements

We are grateful to Dr. Carl Sunshine and Professor P.T. Kirstein for some helpful comments during the preparation of this paper.

REFERENCES

Binder 75 : "The Atlantic Satellite Broadcast Experiments" - R.Retburg, D.Walden - BBN Report 3056, April 1975.

CCITT 76 : "Draft Recommendation X-25" -AP V1-No. 55-E, CCITT, 1976.

Cerf 74 : "A Protocol for Packet Network Intercommunication" - V. Cerf, R.Kahn - IEEE Trans on Communications, p.637, May 1974.

Cerf 78a : "Specification of Internet Transmission Control Program - TCP Version 3" - V.Cerf, J.Postel - Information Sciences Institute, Univ. of Southern California, January 1978.

Cerf 78b : "Proposal for an Internetwork End-to-End Transport Protocol" - V.Cerf, A.McKenzie, R.Scantlebury, A.Zimmermann -Computer Network Protocols Symposium, Liege, Belgium, p.H-5, February 1978.

CYCLADES 73 : "Specifications Functionelles des Stations de Transport du Reseau Cyclades Protocols ST-ST" - Reseau Cyclades SCH.502.3, Institut de Recherche D'Informatique et D'Automatique, Rocquencourt, France, May 1973.

Dalal 74 : "More on Selecting Sequence Numbers" - Y.Dalal - INWG Protocol Note 4, October 1974.

Danet 76 : "The French Public Packet Switching Service: The Transpac Network" - A.Danet, R.Despres, A.Le Rest, G.Pichon, S.Ritzenhaler - ICCC Proceedings, Toronto, p.251, 1976.

Day 75 : "A Note on Some Unresolved Issues for an End-to-End Protocol" - J.Day - INWG General Note 98, August 1975.

Edge 77 : " Buffer Management Strategies for Communications Network Host Protocols with particular Reference to TCP" - S.W.Edge, A.J.Hinchley - Indra Note 611, Dept. of Statistics & Computer Science, UCL, March 1977.

EIN 76 : "End-to-End Protocol" - EIN Publication EIN/76/003.

Erskine 77 : "Datapac: A Packet Switching Network for Canada" - S.B.Erskine - Online Conference Proceedings, Uxbridge, England, p.25, May 1977.

Gelenbe 78 : "Performance Evaluation of The Protocol HDLC" - E.Gelenbe, J.Labetoulle, G.Pujolle - Computer Network Protocols Symposium, Liege, Belgium, p.G-3, February 1978.

Guilbert 77 : "The Transpac Network" - J.F.Guilbert - Online Conference Proceedings, Uxbridge, England, p.15, May 1977.

Kunzelman 78 : "Overview of the Arpa Packet Radio Experimental Network" - R.C.Kunzelman - IEEE CompCon, San Francisco, p.157, Spring 1978.

Mathis 77 : "Single Connection TCP Specification" - J.Mathis - Packet Radio Network Development, Quarterly Technical Report, Appendix B, February-April 1977.

McKenzie 74 : "Internetwork Host-to-Host Protocol" - A.McKenzie - INWG General Note 74, December 1974.

McQuillan 77 : "The ARPA Network Design Decisions" - J.M.McQuillan, D.C.Walden - Computer Networks, Vol 1, No. 5, p. 243, August 1977.

Metcalfe 73 : "Packet Communication" - R.M.Metcalfe - MAC TR-114, MIT, Project MAC,(Ph.D. Thesis), December 1973.

Metcalfe 76 : "Ethernet: Distributed Packet Switching for Local Computer Networks" - R.M.Metcalfe, D.R.Boggs - Communications of the ACM, vol 19, No. 7, p.395, July 1976.

Pouzin 73 : "Efficiency of Full-Duplex Synchronous Data Link Procedures" - L.Pouzin - INWG General Note 35, 1973.

Pouzin 76 : "Flow Control in Data Networks - Methods and Tools" - L.Pouzin - ICCC Proceedings, Toronto, p.467, 1976.

Reed 77 : "The Initial Connection Mechanism in DSP" - D.P.Reed - Local Network Note 10, MIT Laboratory for Computer Science, August 1977.

Sunshine 75 : "Interprocess Communication Protocols for Computer Networks" - C.A.Sunshine - TR-105, Digital Systems Lab, Stanford University, (Ph.D. Thesis), December 1975.

Sunshine 77a : "Interconnection of Computer Networks" - C.A. Sunshine - Computer Networks Vol 1, No. 3, p.175, January 1977.

Sunshine 77b : "Efficiency of Interprocess Communication Protocols for Computer Networks" - C.A.Sunshine - IEEE Tans on Communications, p.287, February 1977.

Tomlinson 74 : "Selecting Sequence Numbers" (DRAFT) - R.Tomlinson - INWG Protocol Note 3, August 1974.

APPENDIX 1 - Analysis of an Algorithm to Estimate Round Trip Time

The algorithm updates the current round trip delay estimate T_n, for a connection, with successive round trip measurements t_n according to either of the following equivalent expressions:

$$T_{n+1} := (m.T_n + t_n)/(m+1) \qquad m >= o \quad (3)$$

$$= T_n + (t_n - T_n)/(m+1) \qquad (4)$$

If the actual round trip delay changes suddenly, and the latest measurement t_n is a better estimate than T_n, then Eq.4 shows T moves closer to the more correct value for smaller m. Conversely, in steady state, after a sequence $t_O, t_1, \ldots, t_n$ of updates, T_{n+1} is given by (using Eq.3):

$$T_{n+1} = (m+1)^{-1}.(t_n + m(m+1)^{-1}t_{n-1} + m^2(m+1)^2 t_{n-2} + \ldots + m^n(m+1)^{-n}t_O + m^{n+1}(m+1)^{-(n+1)}T_O)$$

If we regard each t_i as a random variable, and make the approximation that they are independent and identically distributed (whence the steady state assumption), then simple expressions for the expectation (E) and variance (var) of T_{n+1} can be obtained for the case n approaches infinity:

$$E(T_{n+1}) = E(t_i) \qquad \text{for each } i$$

$$\text{var}\ (T_{n+1}) = (m+1)^{-2}.(\text{var}(t_n) + m^2(m+1)^{-2}\text{var}(t_{n-1}) + m^4(m+1)^{-4}\text{var}(t_{n-2}) + ..)$$

$$= \text{var}(t_i)/(2m+1) \qquad \text{for each } i$$

The last expression above shows that statistical fluctuation of T_n is reduced for larger m.

APPENDIX 2 - Average Letter Delay with Positive Acknowledgement Retransmission

We use previous analytical study (Sunshine 75 and Sunshine 77b) to calculate the average letter delay for the connection model defined in Section 5. The density function f for letter transit delay (x) is, from Eq.2, Section 5:

$$f(x) = (k.u)^{k}.x^{k-1}.e^{-kux}/(k-1)! \qquad x>=0$$

The probability distribution F is:

$$F(x) = 1 - e^{-kux}.\sum_{j=0}^{k-1}(kux)^{j}/j! \qquad x>=0$$

We now add in the effects, successively, of the subnet loss probability p, the retransmission interval R, and the requirement of sequenced letter delivery. The probability F′ of letter arrival after time x, with loss, is:

$$F'(x) = (1-p).F(x)$$

The probability G of arrival, with retransmissions, is:

$$G(x) = 1 - \prod_{j=0}^{n}(1 - F'(x-jR)) \qquad ; n=\lfloor x/R \rfloor$$

The probability H of arrival with all predecessors present is:

$$H(x) = \prod_{j=0}^{\infty} G(x+jA) \qquad ; A = \text{inter letter arrival time}$$

Average letter delay is:

$$\text{average letter delay} = \int_{0}^{\infty}(1-H(x)).dx$$

VIII. ARPANET USAGE

8.1 Organisational Support, Users and Future Provision of Services

There was a significant change in 1978 in the way ARPANET usage was regarded in the UK. Up to the end of 1977, a grant from the SRC Computer Science Committee had covered User Support for University Users. Since that time, it has been decided that there is no longer a research element in the provision of the service. Instead, the Facilities for Computing Committee of the Science Research Council has authorised an agreement to provide support for the service. This support is, therefore, part of the provision of computing services to UK academic research workers.

However, the Facilities Committee has warned that it may well not support usage after 1979. We are exploring whether alternative communications facilities have yet become available, e.g. via the International Packet Switched Service (IPSS) and Telenet. The EDUNET Community may provide such an alternative vehicle. At the end of 1978, host connection of UK Hosts to IPSS was not yet available. Our own line to IPSS is not expected before April 1979. Moreover, normal X25 network access (which UK University Hosts will use to the UK PSS) is not expected before October 1979. Thus it is improbable that a comprehensive range of services, like File Transfer and widespread local interactive terminal support, would be possible by the end of 1979.

The users outside the academic community, particularly the Atomic Energy Authority Culham Laboratories and several Ministry of Defence groups, continue to desire access. In Culham's case, the access requirement is dependent on the Department of Energy's Livermore Laboratories not being available via any other route. They have indicated that they would like continued access at least up to the end of the first quarter of 1980. The Ministry of Defence is organising much longer term collaborative activities with US research groups. They expect to require some form of ARPANET access well into the 1980s.. The groups authorised at the end of 1978 to use the ARPANET link are listed in Table 8.1.

8.2 Technical Quality of Services

The actual quality of the services provided during 1978 was not as satisfactory as in previous years. The reasons for the poorer quality are discussed below. Measures have been taken to overcome the deficiencies when they are under our control, and a much better service is expected in 1979. A number of unfortunate service interruptions occurred in the second half of 1978. These were due mainly to the effects of industrial disputes in the Post Office, to equipment problems in Norway and to problems with our own air conditioning. The first affected the length of time required to restore service in the event of line failure. The second was caused by several incidents of lightning damaging equipment in Norsar during vacation periods. Both these occurrences caused total isolation from the US for several days at a time. The third forced us to shut down the PDP 9s, but allowed the TIP character terminal service over the switched public telephone service to be maintained.

The new release of PDP 9 operating and network systems in mid-1978 allowed access via EPSS to be extended to an essentially 24 hours a day service. The usage via EPSS on a regular basis has now grown significantly; an "E" in the Usage Table of Table 8.1 indicates access via EPSS. Full scale production access via EPSS brought out some unfortunate mismatches between a line-at-a-time system, like the VPT character support, and some of the character oriented application programs on some ARPANET Hosts. These mismatches caused particular annoyances with multiple echoes of some characters (one locally and one from ARPANET). This partial line forwarding was facilitated by suitable specific action in the UCL PDP 9 Gateway. Annoying consequences could not be removed completely, however; most users with high speed access accepted double echoes as a tolerable way of working. One reason why the quantity of usages via EPSS increased very significantly was the result of the 24 hours a day access avaialbility. Another was the determined effort by many UK Hosts to improve their own EPSS support in preparation for the EPSS Open Day in June - mentioned in Section 4.5.

The provision of password checking and the quality of File Transfer support between the Rutherford Laboratory IBM 360/195s and ARPANET suffered as a consequence of the introduction of the new release of the operating system. The actual operating system has grown so large that the generation of a new version is dependent on the availability of a larger disk than the standard 256 K word disk used for earlier releases. A larger 40 M byte disk was brought into service in 1977. It has not, however, proved very reliable. This has severely hampered system development of the PDP 9s, and delayed our response to fixing software bugs. By the end of 1978 most of the deficiencies had been identified and remedied in our development system. The improvements were scheduled for inclusion in the Service System from the first quarter of 1979. Since further software development planned for the PDP 9s is minimal, we do not anticipate long-term problems in development, but the general level of PDP 9 reliability will clearly worsen.

Finally, the introduction of two developments on ARPANET itself have increased the number of service disconnections. One development is changes in the routing algorithms and other standard ARPANET improvements. The UCL mode of connection by a simple two hop 9.6Kbps spur, is unique on ARPANET. For this reason some faults appeared only when new releases were put in the field. The second is due to interference problems when one tried to put the SATNET circuit as an alternate path. Both problems are expected to be much alleviated in 1979.

TABLE 8.1 APPROVED ARPANET USERS DECEMBER 1978

ORGANISATION	NAME	PROJECT	ACCESS METHOD	SITE
APPLETON LABORATORY (SRC)	M.F. REID	INFRA-RED ASTRONOMY	R	AMES, RL
ARCHAEOLOGY, THE INSTITUTE OF LONDON UNIVERSITY	DR. I. GRAHAM	ALGORITHM DEVELOPMENT FOR ANALYSING ARCHAEOLOGICAL DATA		RUTGERS-10 MIT-MC
ASTON UNIVERSITY	D. AVISON	INVESTIGATION OF THE DATACOMPUTER		CCA
BIRMINGHAM UNIVERSITY	DR. K. LANG	GUIDANCE TO REMOTE USERS OF COMPUTERS		SRI, CMU
BLACKNEST RESEARCH EST.	F. GROVER C. BLAMEY	SEISMOLOGICAL DATA EXCHANGE	R	ISI, CCA, RL LLL
BRISTOL UNIVERSITY	DR. J. ALCOCK	DATA ANALYSIS ON ANTI-PROTON SCATTERING	R	LBL, CMU
BRITISH STEEL CORP.	D. SHORTER	HIGH ORDER LANGUAGE DEVELOPMENT FOR US DOD		ISI
CAMBRIDGE UNIVERSITY	PROF. WILKES	ALGEBRAIC MANIPULATION SYSTEM	E	ISI, MIT, UTAH
CULHAM LABORATORY	R. ENDSOR	ALGOL GENERATOR, ALGEBRAIC SYSTEMS		ILLIAC IV, BBN (for test), ISI
DURHAM UNIVERSITY	DR. F.D. GAULT	EXCHANGE OF HIGH ENERGY DATA & SOFTWARE DEVELOP.	R	LBL, RL
EDINBURGH UNIVERSITY (1)	DR. R. BURSTALL	PROGRAM CORRECTEDNESS, TRANSFORMATION & SYNTHESIS		ISIB, UCLA
EDINBURGH UNIVERSITY (2)	M. GORDON	PROOF GENERATING SYSTEM - LCF		SU-AI
EDINBURGH UNIVERSITY (3)	PROF. D. MICHIE	ARTIFICIAL INTELLIGENCE PROGRAMS		ILLINOIS MYCIN
ESSEX UNIVERSITY	DR. J.M. BRADY	VISUAL INFORMATION PROCESSING	E	MIT-AI
HATFIELD POLYTECHNIC (1)	DR. G.M. BULL	BASIC	E	NBS
HATFIELD POLYTECHNIC (2)	DR. A.V. STOKES	USER INTERFACES ON HETEROGENEOUS COMPUTER NETWORKS	E	USC-ISI & various
IMPERIAL COLLEGE	DR. J. DARLINGTON	PROGRAM SYNTHESIS		SRI-KL

ORGANISATION	NAME	PROJECT	ACCESS METHOD	SITE
KENT UNIVERSITY	DR. T.R. HOPKINS	ODEs, PDEs and PADE APPROXIMATIONS		MIT-MC
KINGS COLLEGE, LONDON UNIVERSITY	N.D. BIRRELL	QUANTUM FIELD THEORY IN CURVED SPACE TIME	R	MIT-MC, RL
LEEDS UNIVERSITY	DR. A.P. McCANN	PORTABLE SOFTWARE	E	NYU
LIVERPOOL UNIVERSITY	P. LENG	ALGOL 68 DEVELOPMENT		CMU
MANCHESTER UNIVERSITY	DR. R.N. IBBETT	MU5 MODELLING ALGOL 68 COMPILER DEVELOPMENT	E	CMU
MEDICAL RESEARCH COUNCIL	DR. R.T. WILKINSON	TELECONFERENCING FOR NEUROPHYSIOLOGY COLLABORATION WITH DONCHIN AT BBN		BBN
MINISTRY OF DEFENCE	D. CURRY	COLLABORATION WITH US ARMY MATERIAL COMMAND HQ		ISI OFFICE-1
NAT. PHYSICAL LAB. (BARBER-EIN)	MRS. J. ARMSTRONG	PACKET SWITCHED NETWORKS PROTOCOLS	E	
NAT. PHYSICAL LAB. (2)	L.A. PINK	INVEST. THE REQUIREMENTS OF UNIVERSAL NETWORK ACCESS LANGUAGE	E	SRI-KL, MIT-AI
NAT. PHYSICAL LAB. (3)	DR. B.A. WICHMAN	PROGRAMMING LANGUAGES (IRONMAN)	E	ISIA
NEWCASTLE UNIVERSITY (1)	A.J. MASCALL	SOFTWARE DEVELOP. TO ACCESS EPSS VIA AN IBM-370 - ASSISTING USERS OF ARPANET	E	various
NEWCASTLE UNIVERSITY (2)	J. EVE	ERROR RECOVERY IN COMPILERS	E	XEROX PARC
NORTH LONDON POLYTECHNIC	DR. A.P. JOHNSON	SYNTHESIS DESIGN OF ORGANIC MOLECULES	R	HARVARD, RL
OPEN UNIVERSITY	DR. M. EISENSTADT	COMPUTATIONAL MODELLING OF JOURNALISTIC TEXTS		ISI
OXFORD UNIVERSITY NUMERICAL ALGORITHMS GROUP (1)	S.J. HAGUE	NUMERICAL SOFTWARE	R	ANL, BBN, RL
OXFORD UNIVERSITY (2)	J.B. MACALLISTER	NUCLEAR PHYSICS	R	HARVARD, ILLINOIS, BBN, RL

ORGANISATION	NAME	PROJECT	ACCESS METHOD	SITE
POST OFFICE	C. BROOMFIELD	SIMP EXPERIMENTS		BBN
QUEEN MARY COLLEGE	J.R. HUTCHINSON	DISTRIBUTED COMPUTER SYSTEMS, INTERNETWORKING	E	XEROX PARC MIT-MULTICS
READING UNIVERSITY (1)	PROF. R.W. HOCKNEY	PARALLEL COMPUTING	R	ILLIAC IV, RL
READING UNIVERSITY (2)	DR. R.H. BERMAN	DYNAMICS OF SPIRAL GALAXIES	R	MIT, RL
READING UNIVERSITY (3)	DR. D. FINCHEM	FUSED SALTS	R	ANL, RL
ROYAL COLLEGE OF ART	DR. P. PURCELL	COMPUTER AIDED DESIGN		HARVARD MIT-MULTIC CMU, UCLA
ROYAL HOLLOWAY COLLEGE	DR. R.M. DAMERELL	USE OF MACSYMA FOR NUMBER THEORY APPLICATIONS		MIT-MC
RSRE (1)	DR. J.M. TAYLOR DR. P.H. MASTERMAN	NETWORKING	E	NELC, MITRE RADC
RSRE (2)	N. NEVE	SUPPORT OF US EVALUATIONS OF CORAL		US NAVY LAB.
RUTHERFORD LABORATORY	MRS. S. WARD		R	RL
SALFORD UNIVERSITY	DR. G. LAWS	SUPERSONIC FLOW PROBLEMS	R	ILLIAC IV, RL
SOUTHAMPTON UNIVERSITY	A.J.G. HEY	GAUGE FIELD THEORIES	R	MIT-MC, RL
SYSTEM RESEARCH LIMITED	DR. G. PASK	CONTRACT WITH US ARMY		
THAMES POLYTECHNIC	T. CROWE	NETWORK EDITORS		SRI
UCL (1)	PROF. P.T. KIRSTEIN	PROBLEMS IN COMPUTER NETWORK DESIGN AND APPLICATIONS	E, R	various, RL
UCL (2)	T. WESTGATE	COMPUTER CONFERENCING		BBN
UCH MEDICAL SCHOOL	DR. L. KOHOUT	EVALUATION OF MEDICAL DATA USING GEDYS PACKAGE		SUMEX-AIM
YORK UNIVERSITY (1)	PROF. I.C. PYLE	CONFERENCE WORD PROCESSING		IDA,USC-ISIC ISIC, ISI
YORK UNIVERSITY (2)	I.D. COTTAM	USE OF MODULA FOR PROGRAMMING SECURE OPERATING SYSTEM KERNALS		

Note: E denotes access via EPSS

R " " " Rutherford Laboratory Network

IX. STANDARDS ACTIVITIES

During 1978 significant work on networking standards has been done by CCITT and ISO, involving national standards groups in all countries aware of the need for standardization. The UCL INDRA group has participated actively in UK deliberations, for instance P.L. Higginson has become rapporteur to two of the ISO working groups on one area of the model.

The thrust of the ISO work has been to develop an architectural model for networking consisting of 7 layers (34). The familiar level to those involved with ARPA TCP working groups is level 4/5- a transport station interface comparable to the interface provided by the TCP protocol.

The model also puts terminal support protocols in a fixed part of the model (level 6), and file transfer protocols are also fitted into the general picture. The aim of the working groups is to take existing defined protocols and modify them to fit the architectural model.

The file transfer protocol-NIFTP is currently being implemented by UCL as described in Chapter 4. Several candidate terminal protocols are being considered by the working groups, and UCL has done a test implementation of one of these.

Work on a transport station interface is an urgent matter for both ISO and CCITT; in order to avoid the need to add a protocol such as TCP, the transport station being designed, assumes that lower layers can both reliably deliver messages and also cope with flow control problems that may arise. Within the UK a candidate protocol has been developed with an INDRA member on the working parties concerned, and UCL has promised to implement and evaluate that protocol, when it has been defined.

The INDRA Group members are involved in the following Network and Standards Committees:

Higginson on PO Study Group 2 - Higher level Protocols and
PO Study Group 3 - Transport Services

Higginson and Kirstein attend different subgroups on

PO Study Group 4 - PSS Transition

FTP Implementors Group

Bennett, Fisher, Frost, Higginson.

UK Euronet AD HOC Host Group

Kirstein is Chairman, Higginson attends.

ISO TC/SC16 (Open System Connection)

Higginson is rapporteur of WG 1 amd WG 2.

BSI DPS 20 (UK counterpart of ISO Committee SC16)

Higginson is on the Committee
Hinchley is on the Model Group WG 1
Bradbury is on the Task Activation Group WG 2
Hinchley is on the Transport Station Group WG 3
Higginson is Chairman of the Terminal Protocol Group WG 4

In addition there are many other specific network groups of the University of London, and the Inter-University Computer Committee.

Related to these activities, two important documents were produced. One (13) investigated in depth possible designs for bulk transfer systems for Euronet. It then defined a Remote Printing Protocol, and considers how the protocol might be implemented. This work is being published by the Commission of the EEC for Euronet use. The second document (35) is the definitive account of a 2 week ISO TC 97/SC16 meeting to define the Presentation Layer of the proposed "ISO layered Model of Protocols for Open Systems Interconnection".

X. COLLABORATORS

The following people collaborated in the work of this report during 1978. A "+" indicates that the appointment was made, an "*" that it ceased, during 1978; individuals carrying both indications changed their status during the year.

Academic Staff : P.L. Higginson, P.T. Kirstein, S.R. Wilbur

Technical Supervisor : A.R. Duncan*, H.R. Gamble^{+}

Research Fellow : A.J. Hinchley

Research Assistants : C.J. Bennett^{+}, C. Bradbury, R.C. Cringle, S. Das, S.W. Treadwell, S. Yilmaz.

Engineer : B. Jones^{+}, H.R. Gamble*.

Research Student : A. Akinpelu, C.J. Bennett*, S.W. Edge, Z.Z. Fisher, J.O. Johnson*, P. Lloyd^{+}.

Visitor : E. Getz*

Secretaries: M. Massey, E.O. Oakley.

XI. PUBLICATIONS

We list below the publications written during 1978. We include also some publications, marked with * , which were mentioned in the previous report but without page numbers because they were "in the press".

1. Bradbury, C. X25 Asymmetries and how to Avoid them. Comp. Comm. Rev. 8, 3, 25-34, 1978.

2. Bennett, C.J. Supporting Transnet Bulk Data Transfer. Proc. Symp. Flow Control in Computer Networks, Paris, 383-404, 1979.

3. Cerf, V.G. and P.T. Kirstein. Issues of Packet Network Interconnection. Proc. IEEE 66, pp. 1386-1408, 1978.

4. Edge, S.W. Comparison of the Hop-by-Hop and Endpoint Approaches to Network Interconnections. Proc. Symp. Flow Control in Computer Networks, Paris, 359-377.

5. Edge, S.W. and A.J. Hinchley. A Survey of End-to-End Retransmission Techniques. Comp. Comm. Rev. 8,4, 1-18, 1978.

6. Grossman, G.R. and A.J. Hinchley. Issues in International Public Data Networks. Proc. USA-Japan Conference, 1979.

7. Higginson, P.L. The Design of Bulk Data Transfer Systems in Heterogeneous Computer Networks. Proc. SEAS 78, Stresa, 1978.

8. Higginson, P.L. The Design of Bulk Transfer Systems for Euronet. To be published by the Commission of the European Community, Luxembourg, 1979.

9. Hinchley, A.J. Some Service Aspects of the X25 Interface. Advanced Study Institute on Interlinking Computer Networks. North Holland 1979.

10. Hinchley, A.J. and C.J. Bennett. Gateways to Computer Networks. Proc. Networkshop 2, Liverpool U. 105-121, 1978.

11. Kent, S.A. A National Facsimile and Text Virtual System, Small System Software, 3, 2-13, 1978.

12. Kirstein, P.T. Choice of Data Communications Media for Transmission of Facsimile Information. Computer Networks, 2, 179-190, 1978.

13. Kirstein, P.T. Some International Developments in Data Services. ASI on Interlinking Computer Networks, North Holland, 1979.

14. Kirstein, P.T. and S. Yilmaz. Facsimile Transmission in Message Processing Systems with Data Management. ICCC, 1978, Kyoto, 717-725, 1978.

15. Sunshine, C. and A.J. Hinchley. Submission (by IFIP) to CCITT. Implications of recommendation X25 and proposed improvements for Public Data Network Interconnection. CCITT, 1978.

REFERENCES

1. Kirstein, P.T. University College London, Annual Report, 1 January 1977 - 31 December 1977. TR 48, 1978.

2. Hopper, A. Local Area Computer Communication Network, Cambridge University Computer Laboratory, 1978.

3. Wilbur, S.R. Issues in Relation to the Modification of the Cambridge Ring, Indra 724, 1979.

4. Bradbury, C. X25 - A Case Study, Indra 660, 1977.

5. Bradbury, C. X25 LAP Program Documentation, Indra 671, 1979.

6. Bradbury, C. Interface Specifications for the X25 Level 2 (HDLC) Modules used in the LSI-11, Indra 682, 1979.

7. Bradbury, C. X25 Frame Level Implementation, Indra 683,1979.

8. Bradbury, C. COMSYS Specification, Indra 712, 1979.

9. Bradbury, C. X25 and Multiple Communication Lines, Indra 665, 1978.

10. Hinchley, A.J. Some Service Aspects of the X25 Interface, Advanced Study Institute in Interlinking Computer Networks, North Holland, 1979.

11. Bradbury, C. Letter to the Editor. Comp. Comm. Rev., 8, 2, 5-6, 1978.

12. Higginson, P.L. and P.T. Kirstein. The UCL Black Box for the Attachment of BLAISE to X25 Networks, Indra 686, 1978.

13. Higginson, P.L. The Design of Bulk Transfer Systems for Euronet. To be published by the Commission of the European Community, Luxembourg, 1979.

14. Bennett, C.J. and P.T. Kirstein. Satnet and the Provision of Transnet Service, Indra 674, 1978.

15. Kirstein, P.T. The Development of the UCL Configuration, Indra 675, 1978.

16. Hinchley, A.J. Initial Proposals for Joint UCL/RSRE Network Activities, Indra 718, 1978.

17. Kirstein, P.T. The UCL Configuration Plans as of February 1979, Indra 740, 1979.

18. Hinchley, A.J. and C.J. Bennett. Gateways to Computer Networks, Proc. Networkshop 2, Liverpool U., 105-121, 1978.

19. Hinchley, A.J. and C. Sunshine. Submission (by IFIP) to CCITT. Implications of recommendation X25 and proposed improvements for Public Data Network Interconnection. CCITT, 1978.

20. Bennett, C.J. Types of Service in the Catenet, Indra 680, 1978.

21. Fisher, Z.Z. Network Independent FTP. An EPSS Process on PDP-9, Indra 702, 1978.

22. Bennett, C.J. Supporting Transnet Bulk Data Transfer, Indra 717, 1978.

23. Higginson, P.L. and Z.Z. Fisher. Experiences with the Initial EPSS Service, EUROCOMP, 1978, London, 581-600, 1978.

24. Treadwell, S.W. Gnome Users Guide, TR 41, 1977.

25. Bennett, C.J. The Gnome Controller, Indra 659, 1978.

26. Treadwell, S.W. Satnet Measurement Results, Indra 745, 1978.

27. Treadwell, S.W. Gnome Database Documentation, Indra 748,1978.

28. Kirstein, P.T. Facsimile Techniques for On-line Computer Networks, TR 54, 1979.

29. Kirstein, P.T. and S. Yilmaz. Facsimile Transmission in Message Processing Systems with Data Management, ICCC 1978, Kyoto, 717-725, 1978.

30. Yilmaz, S. and P.T. Kirstein. UCL Experiments in Facsimile Transmission using Data Base Management Facilities on Arpanet. EUROCOMP 78, London 35-49, 1978.

31. Kirstein, P.T. Choice of Data Communication Media for Transmission of Facsimile Information. Computer Networks, 2, 179-190, 1978.

32. Yilmaz, S. Facsimile Techniques in Computerised Message Systems. TR 57, 1978.

33. Akinpelu, A.A. Self-checking Software - an approach to reliable facsimile transnet services, Indra, 725, 1978.

34. Reference model on open systems interconnection, ISO/TC97/SC16/N117., ISO, Paris, 1978.

35. Higginson, P.L. Report of the Ad Hoc Group on the "Presentation Control Layer" of ISO/TC97/SC16/WG2, Indra 723, 1978.